

***Títol: Disseny, construcció i programació d'una plataforma  
robòtica de manipulació coordinada d'instrumental quirúrgic***

***Volum: 1***

***Alumne: Joaquim Barrera Sanson***

***Director/Ponent: Àlicia Casals Gelpí***

***Departament: ESAII***

***Data: 21 de juny de 2012***



---

## DADES DEL PROJECTE

*Títol del Projecte: Disseny, construcció i programació d'una plataforma robòtica de manipulació coordinada d'instrumental quirúrgic*

*Nom de l'estudiant: Joaquim Barrera Sanson*

*Titulació: Enginyeria Informàtica*

*Crèdits: 37,5*

*Director/Ponent: Alícia Casals Gelpí*

*Departament: ESAII*

---

## MEMBRES DEL TRIBUNAL *(nom i signatura)*

*President: Josep Fernández Ruzafa*

*Vocal: Pilar Nivela Alos*

*Directora: Alícia Casals Gelpí*

---

## QUALIFICACIÓ

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---



M'agradaria començar agraint a la que ha estat la meva directora durant el transcurs d'aquest Projecte, l'Àlícia Casals. Així mateix, també agraeixo l'ajuda i els coneixements que m'han aportat tant el Jordi Campos com el Joan Basomba. Sense la seva ajuda aquest projecte no hagués estat possible.

Dono les gràcies també a tots aquells que m'han fet costat durant els anys que he cursat els estudis d'Enginyeria Informàtica i durant el transcurs del Projecte Final de Carrera, resultat del qual teniu a les vostres mans.

En especial vull donar les gràcies a la Zaira, l'Anna, el Sergio, la Xesca, el Ricard i a tothom qui, en un moment o altre, ha tingut paciència amb mi.

Gràcies.



# 1. Índex

1. Índex.....	7
2. Introducció .....	10
3. Estat de l'art.....	14
3.1. Tecnologia de Penelope.....	15
3.2. Altres projectes.....	16
4. Definició dels objectius .....	17
4.1. Construcció de la plataforma.....	18
4.2. Anàlisis i construcció d'un segon robot .....	18
4.3. Programació del robot. El firmware. ....	18
4.4. Programació d'una aplicació basada en PC.....	20
4.5. Interacció usuari-sistema .....	20
5. Descripció de les necessitats i les especificacions tècniques.....	22
6. Descripció del sistema.....	24
6.1. Descripció global del sistema.....	24
6.1.1. Base de la plataforma de proves .....	26
6.1.2. Els robots .....	26
6.1.3. L'ordinador .....	28
6.2. Descripció de cada component del sistema .....	29
6.2.1. Braços robòtics .....	29
6.2.1.1. Morfologia i descripció dels robots.....	29
6.2.1.2. La controladora CM5 .....	32
6.2.1.3. Mòdul de comunicació ZigBee.....	33
6.2.1.4. Càmera sense fils .....	34

6.2.1.5.	Base de la plataforma .....	37
6.2.2.	Descripció de l'ordinador de sobretaula .....	38
6.2.2.1.	Receptor sense fils RX-2400 .....	39
6.2.2.2.	Zig2Serial i Serial2USB .....	39
7.	Programació del robot. El firmware.....	42
7.1.	Abstracció del hardware cap al software. Del WriteByte al setRobot. .	42
7.2	Instruccions de posicionament bloquejants .....	49
7.3	Comunicació bidireccional sense fils entre els robots i el PC de suport	50
7.3.1	Objectius de la implementació del sistema de comunicació .....	53
7.3.2	Implementació de les comunicacions sense fils.....	54
7.3.2.1	Aproximació #1 - Configurar el mòdul ZigBee manualment abans d'enviar cada paquet.....	54
7.3.2.2	Aproximació #2 - ZigBee configurat en broadcast. Un paquet de capçalera i un paquet de dades. ....	55
7.3.2.3	Aproximació #3 - ZigBee configurat en broadcast. Un paquet de capçalera i N paquets de dades.....	56
7.3.2.4	Aproximació #4 - ZigBee configurat en broadcast. Un paquet.....	57
7.3.3	Proves de rendiment.....	57
7.3.4	Conclusions .....	59
7.4	Control de pressió durant el tancament de la pinça del robot .....	60
7.5	Unificació del codi. Un codi per N robots.....	61
7.5.1	Inicialització del sistema.....	62
7.6	Modes de funcionament del sistema .....	66
7.6.1	Mode de moviment teleoperat.....	66
7.6.2	Mode de posicionament gestual del robot.....	66



7.6.3	Mode de cerca i tria d'instrumental quirúrgic.....	67
7.6.4	Mode per ordenar instrumental mèdic a la taula d'instrumental .....	69
7.7	Tecnologies i programari utilitzats .....	71
8	Programació de l'aplicació basada en PC .....	72
8.1	Pas de la codificació RGB a HSI.....	75
8.2	Binarització de les imatges.....	77
8.3	Algorisme de processat d'imatge .....	79
8.4	Interfície gràfica d'usuari .....	84
9	Conclusions .....	87
10	Anàlisi del cost econòmic .....	89
11	Treball futur .....	91
11.1	Sistema de visió per computador .....	91
11.2	Braços robòtics .....	92
11.3	Interfície Gràfica d'Usuari.....	92
12	Bibliografia.....	94

## 2. Introducció

---

La informàtica és una ciència que té moltes branques, tant d'estudi com d'aplicació. Entre elles se'n poden destacar Arquitectura de Computadors, Enginyeria de Software o Llenguatges. Actualment, la Robòtica es considera també una branca més de la informàtica; es tracta potser de la branca més multidisciplinària possible, ja que en un projecte de robòtica es poden dur a terme tasques que van des de la mecànica i l'electrònica fins a la informàtica purament dita.

Això està motivat clarament pels orígens de la robòtica moderna, a mitjans del s.XX, quan els enginyers eren els encarregats del disseny, construcció i programació del robot; és a dir, controlaven tot el procés de fabricació. Es tractava de robots senzills i simplificats, sense funcionalitats avançades de comportament humà, equilibri, etc. Aquest fet capacitava a qualsevol persona amb uns coneixements mínims en robòtica tant a introduir noves modificacions, com a pensar en nous dissenys i programar-los.

Amb els anys la robòtica va creixent i es va fent cada cop més complexa. El treball que fins ara podia dur a terme una sola persona -especificacions, disseny i programació-, ara necessita enginyers i especialistes per al seu desenvolupament. Apareixen nous materials que permeten una fiabilitat més elevada, nous elements actuadors –p.ex. pinces més precises, actuadors magnètics i, en general, més difícils de calibrar-, nous sensors que permeten fer arribar al robot informació de tota mena, i finalment el que marca la diferència del passat al present de la robòtica: la miniaturització de tots els components electrònics, que permet incorporar molta més capacitat de càlcul en un espai molt reduït.

Aquest fet catalitza la introducció de la informàtica en molts camps on abans havia estat ignorada, senzillament per la dificultat de la seva integració i el cost econòmic derivat. S'obren les portes, doncs, a la informatització dels processos industrials, de l'automoció, la recerca aeroespacial, etc. Sense anar més lluny,

un dels camps més coneguts on la informàtica hi ha tingut un paper cabdal ha estat, sense cap mena de dubte, l'astronàutica. Qui es pot plantejar aquesta pràctica sense els ordinadors de bord?

A mesura que es va investigant i miniaturitzant la informàtica, es va introduint en molts àmbits diferents. Un d'ells, especialment el que ens ocupa en aquest projecte, és el camp de la medicina. Sembla evident que la precisió amb la que treballa una computadora, la quantitat de càlculs per segon que pot dur a terme, i els avantatges dels models humans en 3 dimensions, entre d'altres, són factors clau a l'hora de plantejar la integració dels ordinadors i la robòtica a quiròfan.

Estem, doncs, davant del repte de fer que la informàtica aplicada juntament amb la robòtica esdevingui la clau de la medicina moderna, fent possibles investigacions i intervencions abans inimaginables degut a les limitacions pròpies de les persones com la precisió, capacitat de càlcul, memòria, coordinació de múltiples graus de llibertat simultàniament, etc. Un exemple clar de com ja es fa servir la robòtica el trobem fàcilment a les sessions de radioteràpia on, un braç robòtic i no pas cap humà, és qui poc focalitzar en cada instant el punt a irradiar al pacient i controlar la quantitat de radiació que porta acumulada en cada part del seu cos. Aquesta teràpia seria impensable dur-la a terme sense l'ajut de la robòtica.

Segurament l'exemple més conegut dins del camp de la medicina i de la cirurgia és el robot Da Vinci<sup>1</sup>. Aquest robot teleoperat permet desenvolupar intervencions amb gran precisió, amb millors mesures de seguretat i, com en la cirurgia laparoscòpica, amb uns temps de recuperació molt inferiors que no pas amb l'ús de la cirurgia convencional oberta. Tot i així, encara hi ha procediments que es duen a terme de forma més eficient per humans.

La robòtica ha mostrat ja els seus avantatges en molts aspectes, encara que n'hi ha també a millorar. A Catalunya hi ha quatre unitats funcionals del Da

---

<sup>1</sup> <http://www.davincisurgery.com/>

Vinci, a l'hospital de la Vall d'Hebron, a la Clínica Puigvert, a l'hospital de Bellvitge i a la clínica Quirón. La clínica Teknon va anunciar al novembre del 2011 la seva intenció d'adquirir una unitat Da Vinci<sup>2</sup>.



Figura 1 – Sistema Da Vinci de teleoperació quirúrgica.

Nosaltres no centrarem el nostre projecte en el robot orientat a l'execució de la pròpia intervenció quirúrgica. El centrarem en les tasques fins ara reservades exclusivament per als humans, en part degut a la nostra capacitat de raonament, de visió ocular i de comprensió del llenguatge humà. Aquestes tasques són les d'assistència al cirurgià durant el transcurs de la intervenció i la de gestió de l'instrumental mèdic estèril en un hospital; esdevenir el perfecte company del doctor o doctora que condueix una operació és una tasca que solament els assistents més ben capacitats poden dur a terme, ja que requereix d'un coneixement de l'entorn i de les eines de treball molt ben assimilats. Així mateix, portar el control i el registre dels estris que s'han fet servir, dels que han

<sup>2</sup> [http://www.teknon.es/ca\\_ES/noticias/-/asset\\_publisher/76sO/content/noticia-teknon-invierte-en-tecnologia-con-el-robot-da-vinci](http://www.teknon.es/ca_ES/noticias/-/asset_publisher/76sO/content/noticia-teknon-invierte-en-tecnologia-con-el-robot-da-vinci)

estat esterilitzats, ordenar-los a la taula d'assistència, etc. també és una tasca reservada fins ara als humans.

Estem davant el repte de construir un sistema que haurà d'oferir solucions als problemes amb que es pot trobar el personal assistent en un quiròfan. Aquest sistema vol ser una eina amb la precisió d'un robot i amb la potència d'un ordinador, al servei de la salut i la qualitat de la medicina.

### 3. Estat de l'art

---

Si bé a l'apartat introductori s'ha fet esment del robot Da Vinci per les seves qualitats com a assistent al cirurgià, en aquest apartat ens cenyirem a descriure l'estat de l'art sobre la robòtica i la informàtica aplicades a l'assistència en el procés de cirurgia i, en especial, a la gestió d'instrumental estèril.

En particular descriurem els robots *PenelopeCS*<sup>3</sup> i *Penelope*, desenvolupats per l'empresa *Robotic Systems & Technologies Inc.*

El primer d'aquests dos sistemes, *PenelopeCS*, es compon d'un braç robòtic dissenyat amb la finalitat de gestionar el subministrament de material estèril dins d'un quiròfan. Els processos que comporten aquesta tasca, sovint repetitius i on els humans són susceptibles de cometre errades, són l'entorn ideal per a introduir un robot. Incansable i sempre precís, *PenelopeCS* és capaç de dur a terme tasques com:

- Diferenciar entre 3.000 tipus d'eines.
- Localitzar i identificar components mitjançant visió per computador.
- Comptar gases/lleçons.
- Ordenar eines.
- Programar esdeveniments.
- Generar informes d'inventari.

Previ al desenvolupament de *PenelopeCS* existeix el que l'empresa anomena simplement *Penelope*, un braç robòtic que assisteix en les intervencions quirúrgiques. Amb capacitats com el reconeixement de comandes per veu, una capa software d'Intel·ligència Artificial que gestiona el número d'eines utilitzades al sistema, i la possibilitat de calcular la probabilitat de quin instrument serà utilitzat en un futur, entre d'altres, es tracta d'un robot eficient i capaç d'assistir de forma autònoma en un quiròfan.

---

<sup>3</sup> <http://www.roboticsystech.com/products/penelopeCS.html>

La primera intervenció en la qual va participar *Penelope* va ser el 16 de juny de l'any 2005, al NewYork-Presbyterian Hospital<sup>4</sup>. Degut a l'execució correcta de totes les ordres de veu que se li van donar, se'n va fer una valoració positiva.



Figura 2 – Imatge del sistema Penelope per manipular instrumental mèdic.

### 3.1. Tecnologia de Penelope

Tot i que, com és previsible, l'empresa fabricant no dóna tots els detalls que ens agradaria, a través dels vídeos d'exemple que existeixen a Internet<sup>5</sup> podem extreure algunes conclusions:

- Existeix un sistema de visió per computador extern al robot. Aquest sistema està compost per un mínim de dues càmeres situades al sostre de la sala. Amb aquestes càmeres es pot identificar i tenir localitzades totes i cada una de les eines que hi ha a la taula, així com guiar i corregir el braç robòtic durant els seus moviments. Al disposar d'un mínim de dues càmeres, el sistema té la capacitat d'executar la tècnica d'estereovisió<sup>6</sup>, permetent així determinar la profunditat de camp de la imatge i poder treballar en espai tridimensional.
- El braç robòtic és de gama alta i qualitat industrial, permetent una major precisió i fiabilitat en el moviment del robot.

<sup>4</sup> <http://www.roboticsystech.com/infoarticle22presspressRelease22.html>

<sup>5</sup> <http://www.roboticsystech.com/moviesmoviepenelopeCSTour.html>

<sup>6</sup> <http://cabrillo.lsi.uned.es:8080/aepia/Uploads/27/273.pdf>

### 3.2. *Altres projectes*

A casa nostra també s'han dut a terme projectes que anaven en la línia de l'assistència robòtica en cirurgia. En particular, trobem un projecte titulat '*Sistema robòtic per assistència puntual en quiròfans*', desenvolupat en el marc del Màster en Enginyeria Biomèdica per l'alumne Luis Botaya, el 2011.

Els objectius principals d'aquest projecte de màster eren "*desenvolupar un prototip que fos capaç de fer el seguiment d'un objecte, el posicionament en punts determinats de l'entorn i la teleoperació del robot, amb la finalitat d'avaluar la seva contribució a la millora de les condicions de treball en entorns quirúrgics*".

El resultat d'aquest projecte va ser la construcció d'un braç robòtic utilitzant les peces d'un kit de desenvolupament de l'empresa Bioloid anomenat '*Bioloid Comprehensive Robot Kit*'<sup>7</sup> –d'ara en endavant 'kit de desenvolupament de Bioloid'. Aquest robot va assolir els objectius que perseguia el projecte i va posar les bases per al projecte que ens ocupa actualment.

---

<sup>7</sup> <http://www.robotshop.com/robotis-bioloid-comprehensive-kit-3.html>



## 4. Definició dels objectius

---

Un cop feta la contextualització necessària i analitzats els diferents sistemes que hi ha actualment, es procedirà a definir els objectius concrets del projecte que ens ocupa.

L'objectiu principal i l'entorn en el qual es desenvoluparà l'activitat del projecte, és el disseny, la construcció i la programació d'una plataforma robòtica que permeti l'assistència coordinada i la manipulació d'instrumental quirúrgic abans i durant una intervenció.

Aquesta plataforma, que estarà composta per dos braços robòtics i integrada en un context de quiròfan, ha de ser capaç d'assistir a l'equip de cirurgia que es trobi intervenint en aquell moment, assumint tasques com l'extracció de material de rebuig, l'acostament i la subjecció d'eines de treball i la detecció de situacions de risc. A més a més, també ha de ser capaç d'assumir tasques de gestió del material estèril, com ara el recompte i l'ordenació previs i posteriors a una intervenció, entre d'altres.

La coordinació entre els dos braços robòtics de la plataforma ha de fer possible que el seu funcionament estigui deslligat de la interacció amb cap ordinador extern, exceptuant el processat d'imatge, que per motius de potència de càlcul passarà irremediablement per un PC.

Un cop la plataforma sigui operativa, els braços robòtics seran capaços de funcionar de forma coordinada -com si es tractés d'un de sol- més que no pas com a dos braços independents.

Tot seguit es presenten breument els objectius en els quals s'ha treballat durant el projecte. En els apartats següents es descriurà el sistema que s'ha construït i s'explicaran tots i cada un dels detalls rellevants del seu disseny, les decisions que s'han pres, etc.

#### **4.1. Construcció de la plataforma**

L'objectiu és dotar a la plataforma robòtica de les característiques que es poden trobar a la realitat, dins d'un quiròfan. Aquesta plataforma ha d'oferir la possibilitat de ser utilitzada com a llit d'operacions o com a taula d'instrumental, ja que ens ha de permetre desenvolupar ambdues tasques: la d'assistir en una operació i la de gestionar material estèril.

A més a més ha de tenir la capacitat d'albergar i fixar la base dels dos robots, de forma que collar-los sigui senzill. Per últim, també cerquem versatilitat i la possibilitat de que pugui ser transportada i muntada sense molta dificultat allà on faci falta.

#### **4.2. Anàlisis i construcció d'un segon robot**

Partint del primer robot ja implementat al projecte del Luis Botaya, i per dotar al sistema de la funció col·laborativa, és necessària la construcció d'una segona unitat robòtica, per tal de duplicar l'espai de treball global del sistema i oferir una assistència més completa.

S'ha de fer una anàlisis de com està construït el robot del projecte comentat a l'Estat de l'Art, per veure si es poden introduir millores i, finalment, replicar-lo en una segona unitat robòtica.

#### **4.3. Programació del robot. El firmware.**

En el codi del robot, és a dir, en la programació del sistema, és on realment es marca la diferència qualitativa. Quan es va començar a desenvolupar el codi es tenia clar que el que es faria seria un sistema que fos:

- Escalable, de forma que en un futur no es necessités replantejar-ho tot si es vol ampliar el joc d'instruccions, el nombre de robots, etc.

- Abstret dels detalls característics que conformen un robot. La programació del sistema estarà deslligada d'un coneixement ampli sobre robòtica i cinemàtica –tot i que és molt recomanable-, ja que s'ha dissenyat i implementat un joc d'instruccions fàcil de comprendre per l'usuari, i aquest no hagi de preocupar-se de les articulacions que conformen el robot, els motors, ni detalls de baix nivell. En aquest punt s'ha dedicat especial interès, ja que fer-ho bé comporta que el sistema el puguin programar tant estudiants en pràctiques, enginyers o doctors.
- Intel·ligent, el màxim possible. Partint de la base de hardware a baix nivell que no té controladors associats, l'objectiu és dotar al sistema d'una intel·ligència que el permeti identificar instrumental, saber quan la pinça ha agafat alguna cosa, controlar les eines que hi ha a la plataforma de proves, etc. Totes aquestes petites funcionalitats són les que, posades totes juntes, donen el valor afegit al sistema final.
- Sòlid, intentant minimitzar la probabilitat d'error, fent comprovacions de que les dades són correctes i assumint que sempre hi ha quelcom fora del nostre control. D'aquesta manera, en cas d'error, el sistema es pot recuperar més fàcilment o, com a mínim, causar la menor quantitat de problemes.
- Comprensible, perquè es pugui seguir el fil d'execució del codi sense haver de cercar la definició de les subrutines, ni haver de ser un expert en la programació de microcontroladors. Aquest aspecte va molt lligat a l'estil de programació, i s'ha intentat que tant el nom de les funcions, l'esquema que segueix el codi, la gestió de les rutines de servei, els comentaris i el nom de les variables siguin el més uniformes possible.

#### **4.4. Programació d'una aplicació basada en PC.**

L'objectiu que persegueix el desenvolupament d'una aplicació basada en PC és gestionar el processat de les imatges provinents de la càmera instal·lada al sistema, dotant-lo d'una visió global del seu entorn. Aquest fet persegueix l'objectiu de diferenciar colors i, amb l'instrumental mèdic degudament etiquetat, diferenciar entre una eina i una altra. Tot aquest procés s'ha de poder efectuar en temps real.

A més a més, l'aplicació basada en PC cerca proporcionar una interfície d'usuari entre el cirurgià i el sistema. Al següent punt es descriu aquest objectiu.

#### **4.5. Interacció usuari-sistema**

Finalment, desenvolupar un sistema d'interacció entre l'usuari (entenent usuari com a l'equip de cirurgia) i el sistema robòtic que permeti donar ordres als robots, així com obtenir informació de l'estat del sistema, és un altre dels objectius d'aquest projecte.

Recordem que de la manera com s'ha plantejat, el sistema té com a objectiu ser el més autònom possible, minimitzant la interacció amb l'usuari, per tal d'oferir una assistència intel·ligent. Tot i així, l'equip de cirurgia o d'assistència ha de tenir la possibilitat d'introduir ordres al sistema. Seguint aquesta indicació, el sistema oferirà la possibilitat de ser interactuat a partir de la botonera de la controladora dels robot –sense fer ús de cap ordinador- o bé a través d'una interfície d'usuari programada i executada a l'ordinador central. A la figura 3 es pot observar l'esquema bàsic de com s'ha concebut la interacció usuari-sistema.

Cal observar que de moment no s'especifica cap tecnologia concreta, ja que això es reserva pels apartats següents, on s'especificaran els detalls i limitacions tècniques del sistema en qüestió.

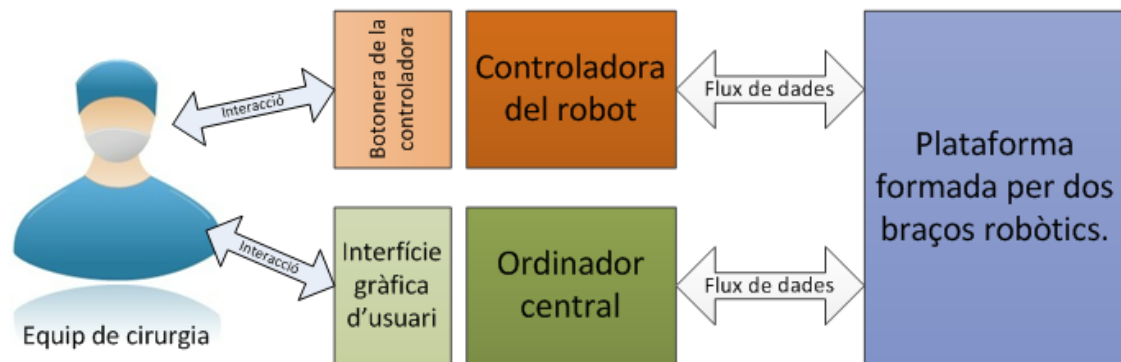


Figura 3 – Diagrama de la interacció usuari-sistema.

## 5. Descripció de les necessitats i les especificacions tècniques

---

Partint dels objectius que s'han descrit anteriorment, aquest projecte comporta tota una sèrie de necessitats, reptes i aspectes que cal abordar, i que tot seguit s'expliquen.

En primer lloc, el sistema ha de poder ser ubicat o instal·lat amb facilitat als voltants d'un llit d'operacions, per tal de donar servei al cirurgià i/o al personal assistent de la intervenció.

Seguidament, i degut principalment a les limitacions d'espai d'una sala d'operacions i a les limitacions de moviment del cirurgià en el moment de la intervenció, es farà un disseny de dos braços robòtics que. Aquest disseny ha de comptar que dins de l'espai de treball individual de cada un dels robots, hi haurà d'haver una zona d'intersecció entre els dos espais de treball, per tal de fer efectiva la transferència d'instrumental i duplicar així l'espai de treball total sistema.

A més a més es plantegen dos aspectes importants que cal abordar.

El primer tracta sobre la identificació de l'instrumental quirúrgic existent a la sala d'operacions. Inicialment s'ha plantejat l'ús d'una càmera instal·lada a l'extrem d'un dels robots i marcar les eines de treball amb diferents colors. Més endavant, però, veurem que s'haurà d'acotar i simplificar aquesta solució donades les limitacions tècniques de la càmera.

El segon aspecte tracta sobre la comunicació que s'ha poder establir entre tots els mòduls del sistema que es vol construir –robots i ordinador-. Aquesta comunicació haurà de ser sense fils, principalment per evitar tota la problemàtica derivada de l'ús de cablejat en un entorn quirúrgic, i a més a més, per facilitar i simplificar el muntatge de la plataforma.

Es proposa, doncs, la construcció d'un plataforma de proves amb dos braços robòtics fixats a una base que representi el llit d'operacions o la taula d'instrumental, indistintament.

Aquest sistema ha de ser autònom, és a dir, cada robot ha d'incorporar la seva font d'alimentació i ha de poder ser controlat sense la mediació de cap altre sistema. Aquest fet no descarta la possibilitat de la interacció amb l'usuari ni de dur a terme la teleoperació dels robots en cas de ser necessari.

Per aquesta raó es planteja habilitar una de les botoneres dels robots com a entrada d'ordres i comandes, implementant un menú d'opcions on, tant el cirurgià com l'equip d'assistència, puguin escollir el mode de funcionament del sistema, obtenir les dades disponibles, etc. Es pretén que aquesta entrada d'ordres sigui extensible a l'aplicació basada en PC.

Com s'ha comentat, caldrà la incorporació d'una càmera –juntament amb l'equipament addicional necessari- per poder identificar instrumental, així com la incorporació d'una pinça a l'extrem de cada un dels, per permetre la subjecció i el transport instrumental mèdic. El sistema de visió ha de permetre identificar un nombre suficient, però finit, d'eines de treball que permetin al cirurgià poder escollir sense cap mena de restricció.

Es proposa la utilització d'un ordinador de sobretaula per encarregar-se del processat d'imatge provinent de la càmera. Aquest processat d'imatge serà efectuat en temps real per una aplicació basada en PC que, a més a més, permetrà efectuar altres accions sobre el sistema.

Finalment, es proposa un sistema de comunicació sense fils que creï i connecti una xarxa de dispositius –dos robots i un ordinador- de forma que qualsevol d'aquests dispositius pugui enviar i rebre dades independentment dels altres dos.

Partint d'aquestes necessitats, a l'apartat següent es pot trobar una àmplia descripció del tot el sistema que s'ha dissenyat, així com les especificacions de més baix nivell.

## 6. Descripció del sistema

---

Vistos els objectius del projecte, els reptes plantejats i finalment les especificacions necessàries per a resoldre'ls, tot seguit es farà una descripció del sistema que s'ha dissenyat per a dur a terme totes les tasques que s'han plantejat.

### **6.1.      *Descripció global del sistema***

En primer lloc es presenta un diagrama global del conjunt del sistema (Fig. 4), de forma que el lector pugui ubicar fàcilment tots els actors, des de la plataforma de proves, els robots, el cirurgià/na, l'ordinador de sobretaula, etc.

Aquest esquema pretén introduir els elements que conformen el sistema, però no entrar-hi en el detall de les seves característiques, etc. D'aquesta manera, i a mesura que s'avanci en la lectura d'aquest document, el lector estarà familiaritzat amb l'arquitectura del sistema robòtic, i quan se'n faci una descripció detallada, serà més senzill seguir-la.





25

### **6.1.1. Base de la plataforma de proves**

Degut a que tant els encerts en el seu disseny, així com també les errades, acompanyarien el desenvolupament posterior de tot el projecte, ha estat del tot necessària una bona planificació de la seva construcció.

S'ha decidit que la plataforma sigui el més realista possible, és a dir, que tingui similituds amb l'entorn de treball dins del quiròfan, i que sigui de muntatge i desmuntatge senzill per permetre el seu transport a les instal·lacions mèdiques per a fer proves –en cas de ser necessari–.

La base de la plataforma de proves ha estat construïda principalment amb l'objectiu de dotar als dos robots d'una àrea d'intersecció entre el seu espai de treball per a poder fer l'intercanvi d'instrumental quirúrgic en temps d'execució. S'ha optat per fer un disseny que representi la secció d'un llit d'operacions, en particular d'una mesura de 55 cm d'ample de robot a robot.

### **6.1.2. Els robots**

Es requereix de la construcció d'una segona unitat robòtica partit de la base marcada per la morfologia del robot dissenyat al projecte del Luis Botaya. El motiu principal de la segona unitat robòtica ha estat per tal de duplicar l'espai de treball que ofereix un sistema mono-robot, i per dotar a la nostra plataforma de la capacitat col·laborativa entre dos robots, acomplint així un dels objectius principals.

El fet de disposar de documentació prèvia ha permès analitzar-la i introduir algunes millores en la seva construcció, com per exemple el reforçament de la base del robot, debilitada pel sobrepès mateix del robot, que va estar mal dimensionat inicialment.

A més a més, durant el muntatge de la segona unitat robòtica s'ha incorporat un component addicional disponible al kit de desenvolupament de Bioloid. Es tracta d'un sensor de so i d'infrarojos –IR–, no present a la primera unitat

robòtica. S'ha decidit incorporar-lo per tal que futures ampliacions del sistema ja trobessin aquest hardware muntat i provat. Per aquest motiu s'identifiquen els robots com 'ROBCAM' –robot dotat de càmera- i 'ROBIR' –robot dotat de sensor d'infrarojos- a les figures 4 i 5.

A continuació es pot observar una ampliació de la figura 4 on s'observen més detalls de l'arquitectura del sistema, que permeten aprofundir en el detalls de la plataforma robòtica.

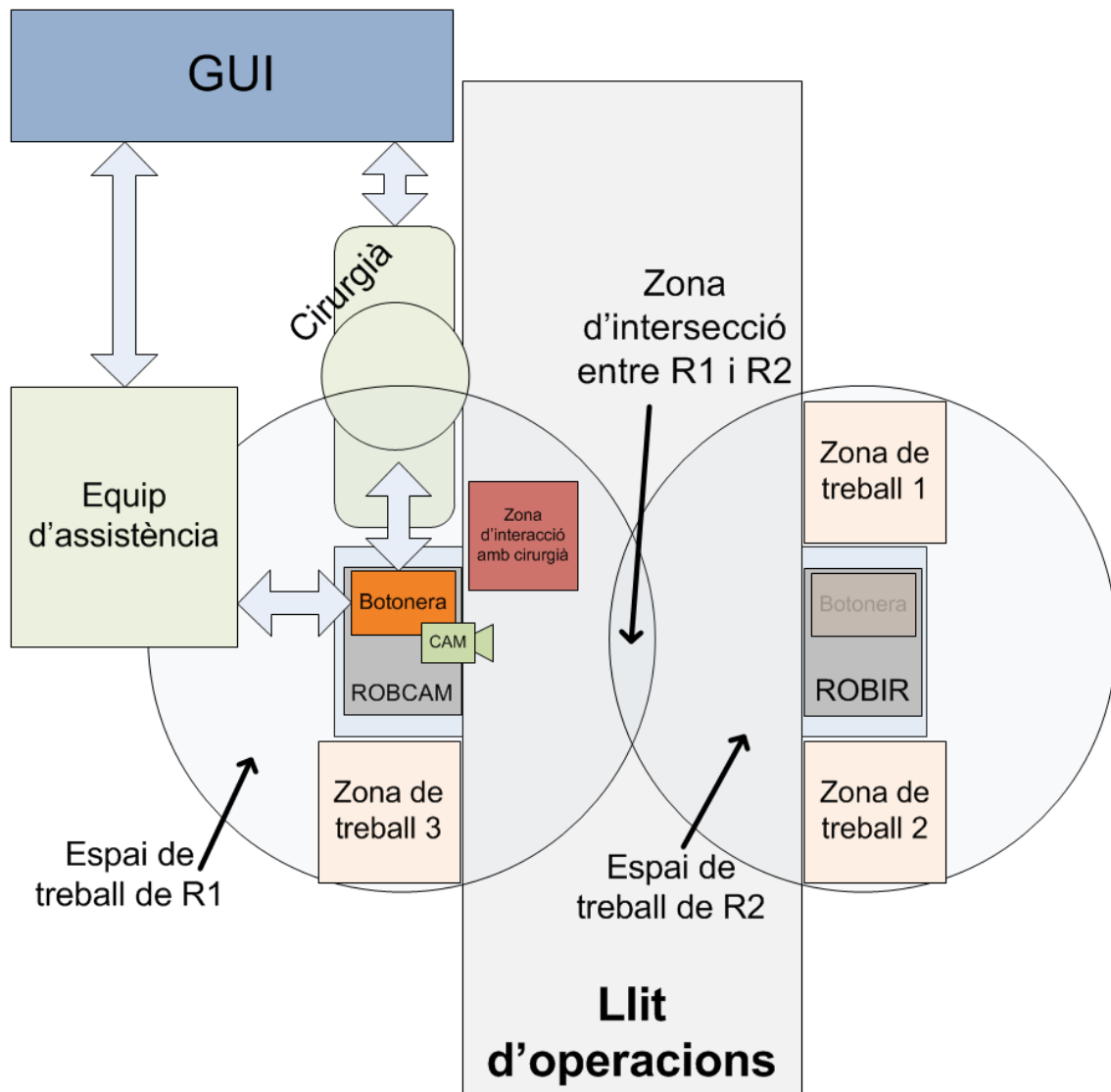


Figura 5 – Esquema d'accessibilitat usuari – sistema, juntament amb les zones de treball.

Cal fixar-se que el sistema sencer (interfície d'usuari (GUI) + botonera del robot) pot ser utilitzat tant pel cirurgià com pel seu equip, indistintament. Això és així degut a que en un moment determinat el cirurgià pot necessitar dipositar tota l'atenció a la taula d'operacions, i delegar en el seu equip que l'assisteix, les tasques de gestió del sistema.

La zona d'intersecció és allà on es troben els dos espais de treball dels robots, i allà on es durà a terme l'intercanvi d'instrumental per moure'l d'una zona de treball a un altre.

Les zones de treball són part de l'espai de treball dels robots on poden accedir i dipositar instrumental, entre d'altres possibles usos.

Per acabar, la zona marcada com d'interacció amb el cirurgià és la zona on es podrà dipositar instrumental per a ser gestionat automàticament pel sistema, tant pel seu rebuig com pel seu retorn a la zona de material.

### **6.1.3. L'ordinador**

Es tracta d'un ordinador, en aquest cas de sobretaula, equipat amb un receptor sense fils per rebre les imatges de la càmera, i un mòdul ZigBee de comunicació sense fils per transmetre i rebre dades dels dos robots. Aquest equip permetrà el processat d'imatge en temps real, enviar i rebre comandes als dos robots, alhora que proporcionarà una interfície d'usuari per a l'equip de cirurgia.

Donat que aquest és un apartat introductori, es trobaran més detalls al punt 6.2.2, on es descriuen cada un dels components que conformen aquest equip.

## **6.2. Descripció de cada component del sistema**

Un cop repassats els components del sistema a nivell general, a continuació es descriuran tots i cada un dels components que conformen la plataforma de proves: els robots i l'ordinador de sobretaula. L'objectiu que es persegueix és, havent donat al lector el coneixement general sobre com està construït el sistema, aportar ara tota la informació tècnica necessària per comprendre el desenvolupament posterior del projecte, així com entendre les justificacions de disseny del software i les limitacions a les quals ens hem enfrontat.

### **6.2.1. Braços robòtics**

En aquest apartat es descriurà tot el que fa referència als dos braços robòtics que formen part d'aquest projecte, des de la seva morfologia fins als components que en formen part.

#### **6.2.1.1. Morfologia i descripció dels robots**

L'arquitectura del robot ROBCAM es mostra a la figura 6. S'observa el robot equipat amb la càmera que permetrà al sistema gaudir de visió per computador, a més a més de les 6 articulacions del sistema i l'eina de treball – en aquest cas una pinça- juntament amb els components més destacables de la controladora, que bàsicament són el mòdul de comunicacions sense fils ZigBee, la interfície de control i la bateria.

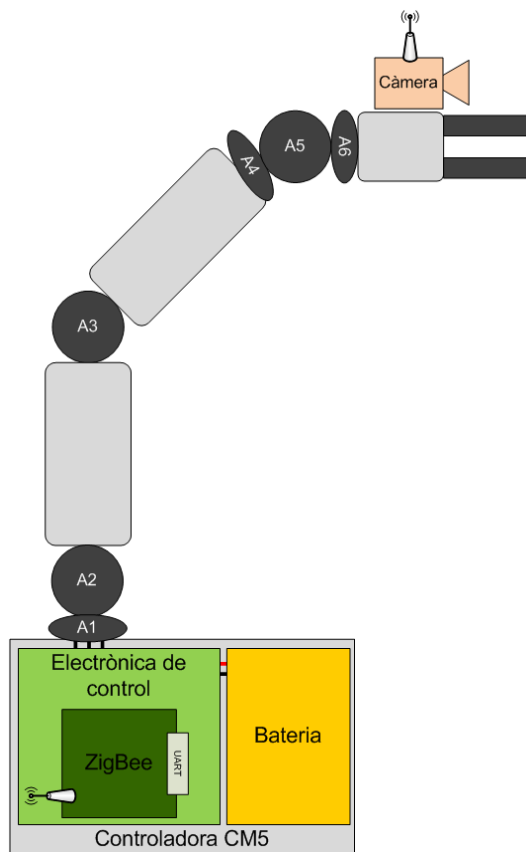


Figura 6 – Robot ROBCAM equipat amb la càmera sense fils.

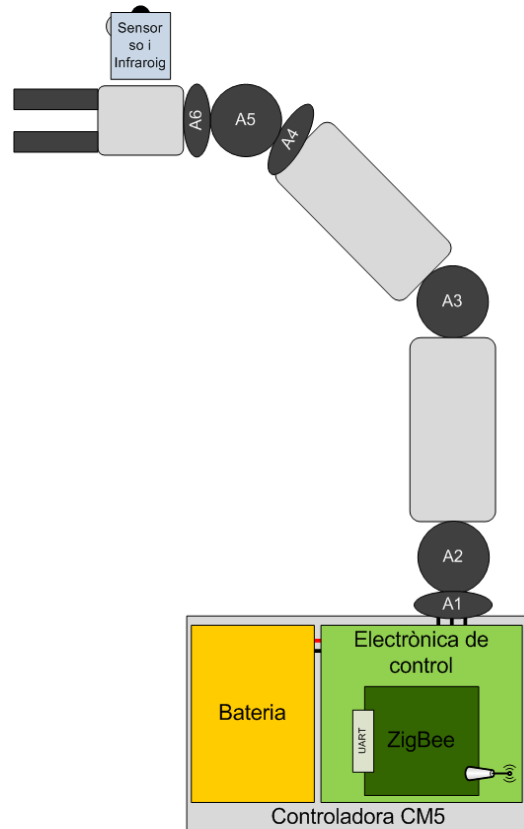


Figura 7 – Robot ROBIT equipat amb el sensor d'infrarojos.

El segon robot és morfològicament idèntic al primer, amb la diferència del sensor que hi ha habilitat a l'extrem; en aquest cas es tracta d'un sensor de so i d'infrarojos. Es pot observar el detall a la figura 7.

Seguint les decisions que es van prendre durant el projecte del Luis Botaya, cada robot està format per nou servomotors de la marca Dynamixel, model AX-12+, distribuïts de la següent manera:

- UN servomotor a la base.
- DOS servomotors a l'espatlla.
- DOS servomotors al colze.
- UN servomotor a la primera articulació del canell.
- UN servomotor a la segona articulació del canell.
- UN servomotor a la tercera articulació del canell.
- UN servomotor per obrir i tancar l'eina del robot.

S'ha mantingut aquesta distribució principalment per dos motius. El primer és que permet una bona mobilitat del robot, ja que els dos motors que conformen l'articulació de l'espatlla (A2) i els altres dos que conformen el colze (A3), donen la potència necessària per permetre moure correctament la resta del robot. El segon dels motius és el fet que al kit de desenvolupament que s'ha fet servir només hi havia 18 servomotors, el nombre just per construir dos robots d'aquestes característiques.

Els 9 servomotors estan connectats entre ells i a la seva controladora mitjançant el bus de comunicacions RS-485<sup>8</sup> on cada element es connecta en sèrie tal i com es mostra a la següent figura:

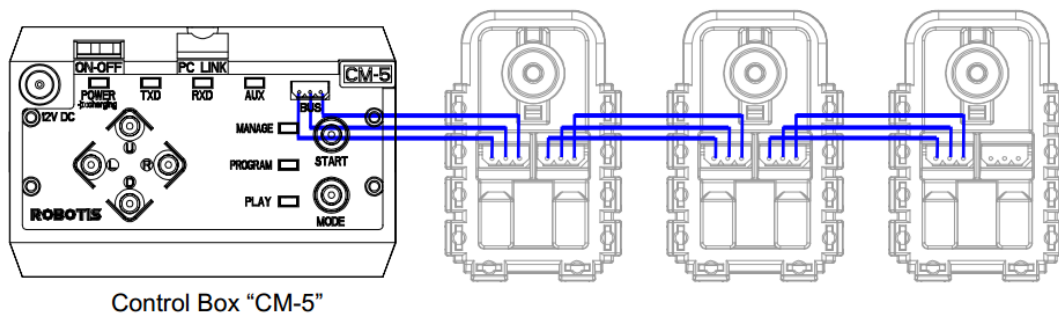


Figura 8 – Esquema de connexió entre els servomotors i la controladora.

Aquesta forma de connexió permet una comunicació amb els motors en forma de paquets identificats per l'adreça de cada motor i les dades associades - similar a com es transmeten les dades al bus I2C<sup>9</sup>-, com es mostra a la següent figura.

<sup>8</sup> <http://en.wikipedia.org/wiki/RS-485>

<sup>9</sup> <http://www.i2c-bus.org/>

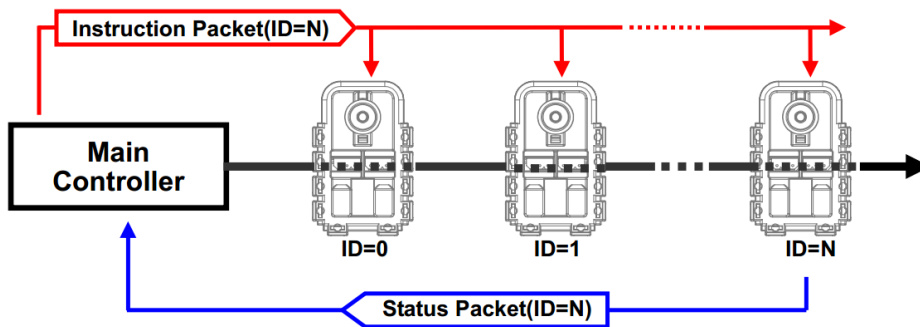


Figura 9 – Esquema de comunicació entre el servomotors i la controladora.

#### 6.2.1.2. La controladora CM5

Avançant en la descripció del sistema, a continuació descriurem la controladora CM5, que és el cervell de cada robot. Integra un microcontrolador ATmega128<sup>10</sup> fabricat per l'empresa Atmel, dotat amb quatre ports sèrie per connectar components del kit de desenvolupament, botons de control programables, LEDs configurables, etc.

La controladora CM5 pot ser ampliada –de fet en aquest projecte ho ha estat– amb la funcionalitat de connexió sense fils, gràcies al mòdul ZigBee, que serà detallat a l'apartat següent.

La controladora s'alimenta amb una bateria de 2300 mAh<sup>11</sup> que, amb un ús mitjà de tots els servomotors, té una vida útil d'aproximadament 15 minuts. Aquest fet limitarà la llibertat del robot i serà un factor a tenir en compte en futurs desenvolupaments.

A la figura següent s'observa un diagrama amb els components principals de la controladora CM5.

<sup>10</sup> <http://www.atmel.com/devices/atmega128.aspx>

<sup>11</sup> <http://en.wikipedia.org/wiki/Ampere-hour>



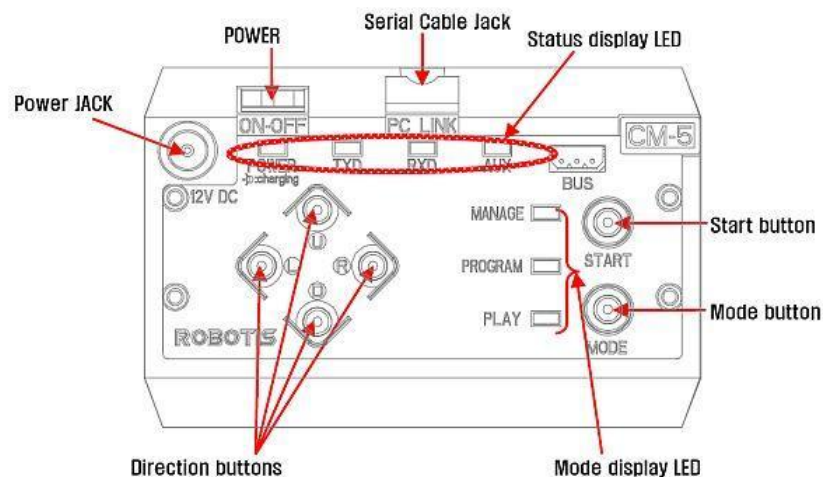


Figura 10 – Vista superior de la controladora CM5.

### 6.2.1.3. Mòdul de comunicació ZigBee

Pel que fa a les comunicacions, degut a les necessitats de que aquestes fossin sense fils, s'ha hagut de cercar una solució que s'adaptés a les nostres necessitats.

Afortunadament aquest cas ja estava contemplat en el kit de desenvolupament de Bioloid, incorporant una unitat del mòdul de comunicacions sense fils ZigBee<sup>12</sup> per a cada controladora, que ha permès dotar tant als robots com a l'ordinador de sobretaula de la capacitat d'interconnexió sense fils. Les característiques principals d'aquest mòdul són:

- Ús de la banda PAN<sup>13</sup> (Personal Area Network) de 2,4 GHz.
- Ample de banda màxim de 250 kbps.
- Interfície de funcionament sèrie *UART*<sup>14, 15</sup>
- Cada mòdul té una adreça pròpia de comunicació de 16 bits.

<sup>12</sup> <http://www.zigbee.org/>

<sup>13</sup> <http://searchmobilecomputing.techtarget.com/definition/personal-area-network>

<sup>14</sup> <http://ca.wikipedia.org/wiki/UART>

<sup>15</sup> [http://www.freebsd.org/doc/en\\_US.ISO8859-1/articles/serial-uart/index.html](http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/index.html)

A les figures 11 i 12 es poden observar els detalls tant de disseny com de connexió del mòdul ZigBee.

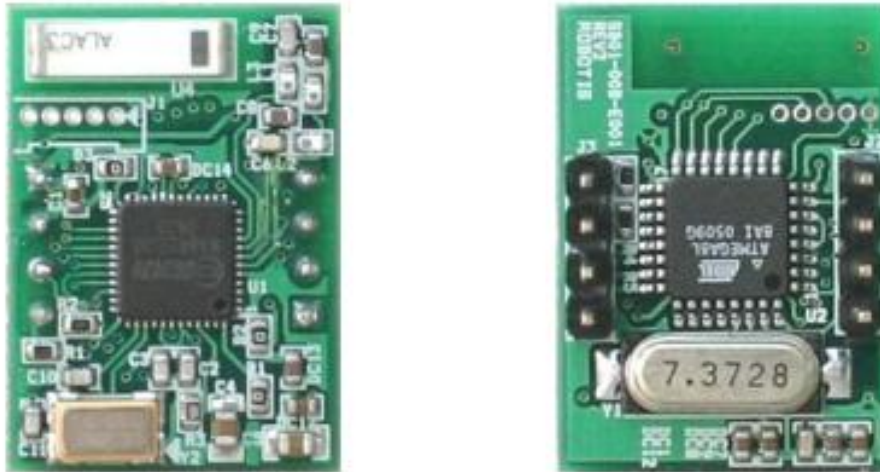


Figura 11 – Anvers i revers del mòdul ZigBee.

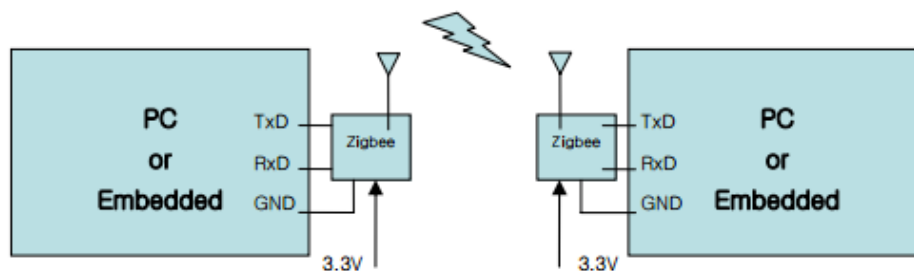


Figura 12 – Esquema de connexió dels mòduls ZigBee.

#### 6.2.1.4. Càmera sense fils

Un altre element destacat del robot és la càmera sense fils. El kit de desenvolupament utilitzat en aquest projecte ja en duia una incorporada, fàcilment instal·lable en un dels robots. Les especificacions tècniques de la càmera són:

- Resolució màxima: 480x320 píxels.
- Quadres per segon: 30.
- Temps de captura: 33,30 ms.
- Temps de processat: 6,20 ms.
- Antena incorporada.

Aquestes dades ens asseguren una imatge interpretable pel sistema de visió per computador que permeti prendre decisions en temps real.

La connexió de la càmera al robot es realitza de la mateixa manera que un servomotor addicional al bus de dades (veure figura 8 per més informació), únicament amb la diferència que la càmera només aprofita l'alimentació del bus, en comptes de les línies de dades.

Aquest fet deriva en que un cop s'inicia el sistema, la càmera comença a emetre la seqüència de vídeo a través d'un circuit sense fils, i aquesta informació és capturada per un receptor connectat a l'ordinador central.

La transmissió d'informació de la càmera al receptor es realitza per un canal diferent del que es fa servir per transmetre dades a través dels mòduls ZigBee.

A la figura 13 es pot observar la càmera instal·lada a l'extrem d'un dels robots.

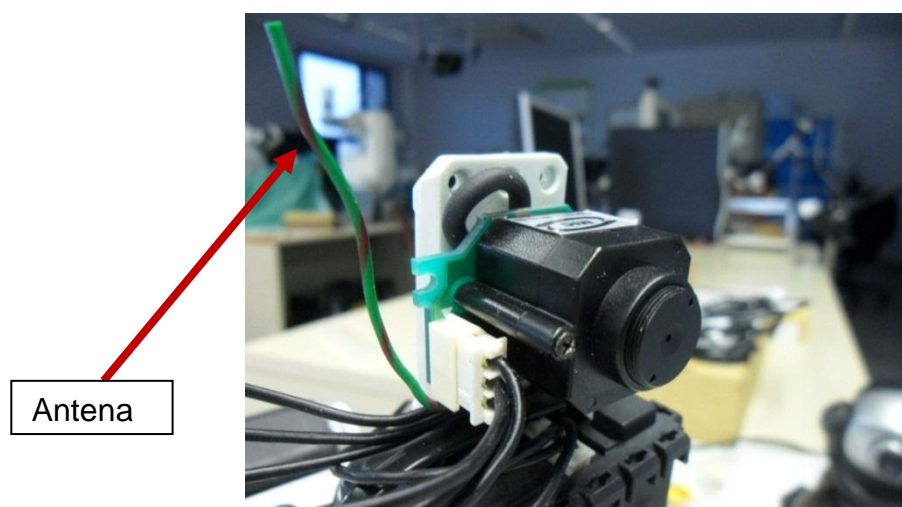


Figura 13 - Càmera sense fils instal·lada a l'extrem d'un dels robots.

El fet de disposar d'una sola càmera no permet l'ús de la tècnica de l'estereovisió<sup>16</sup> per obtenir la profunditat dels objectes de la imatge en temps real.

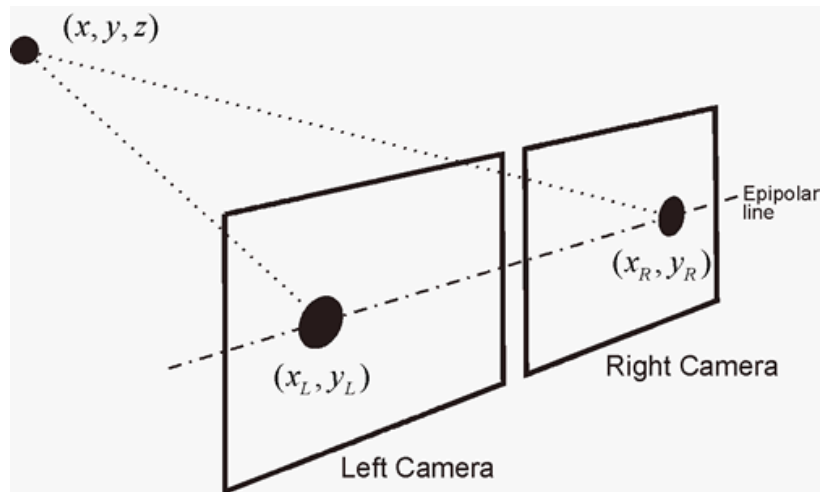


Figura 14 – Ús de dos càmeres per detectar la profunditat del punt  $(x, y, z)$ .

Aquesta mancança es pot solucionar aplicant una tècnica que consisteix en prendre dues imatges del mateix punt ( $(x, y, z)$  a Fig.14) des de posicions diferents, per després aplicar la tècnica de la paral·laxi<sup>17</sup>.

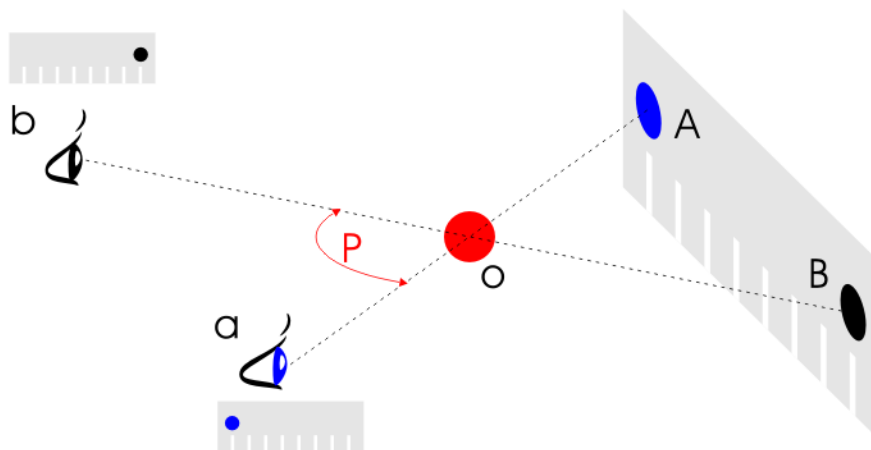


Figura 15 – Il·lustració del mètode de la paral·laxi. Posicions a) i b) des d'on es prenen les imatges de l'objecte O.

<sup>16</sup> <http://www.lsi.upc.edu/~virtual/SGI/guions/RV43.html>

<sup>17</sup> <http://ca.wikipedia.org/wiki/Paral%C2%B7laxi>

Malauradament, per dur a terme aquesta tècnica calen elements de molta precisió i el sistema de posicionament dels robots no ho permeten. Per tant, d'ara en endavant assumirem que les imatges que el nostre sistema de visió per computador tracta són únicament en 2 dimensions.

#### 6.2.1.5. Base de la plataforma

Per completar l'apartat descriptiu del robots, es mostrarà el sistema mecànic de collat a la base de la plataforma. Aquest sistema a pressió s'ha construït a partir d'unes peces de fusta i metacrilat, collades amb uns cargols metàl·lics. A la figura següent es mostra una fotografia del detall d'aquest sistema de collat.

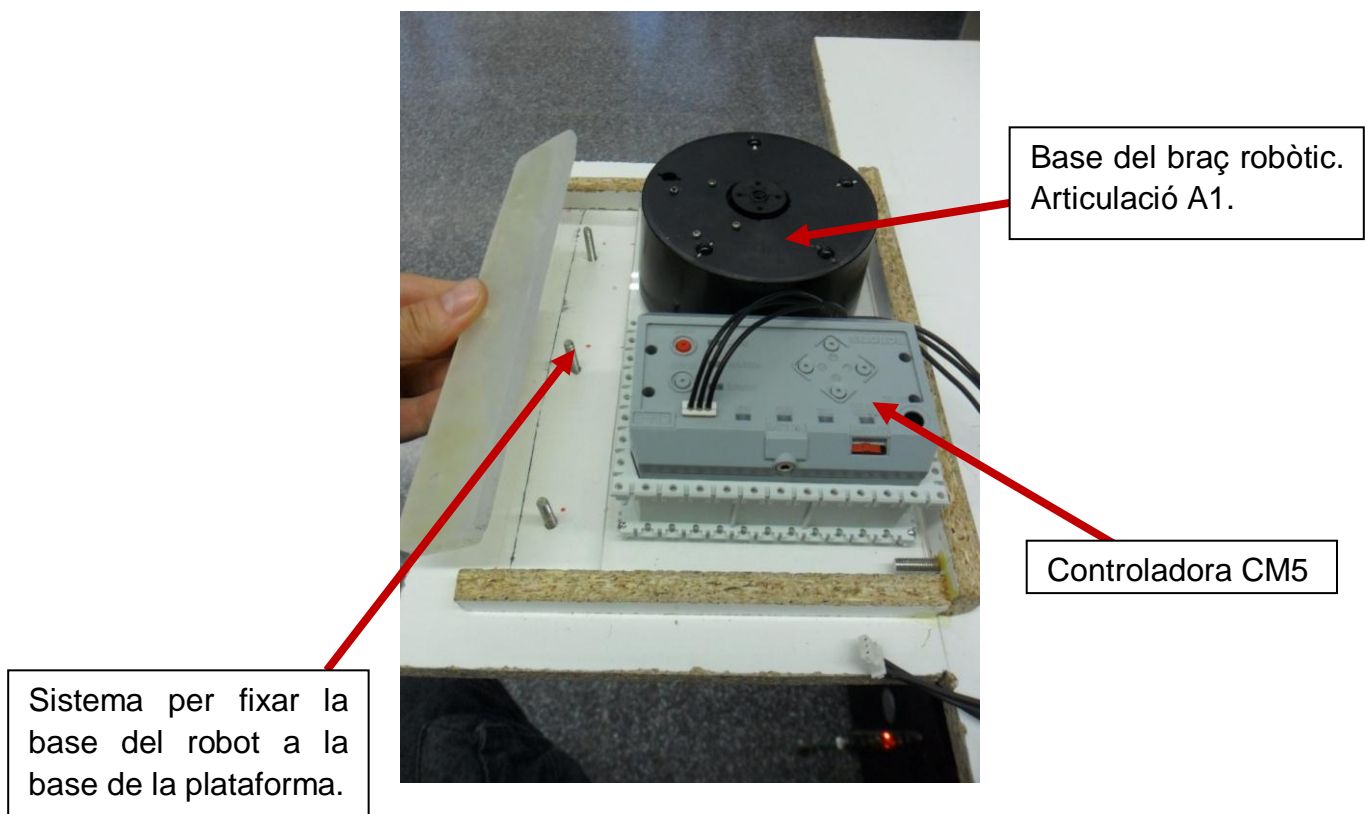


Figura 16 – Sistema de fixació dels robots a la base.

Aquest sistema permet la inserció i el desmuntatge dels robots de manera ràpida i segura, a més d'oferir robustesa i seguretat quan aquests estan funcionant.

### 6.2.2.Descripció de l'ordinador de sobretaula

De la mateixa manera que per començar a descriure els robots s'ha presentat un diagrama del sistema, a la figura 17 es pot observar els elements que intervenen en la connexió de l'ordinador central amb la resta del sistema. Als apartats següents es detalla cada un dels components aquí representats, quines dades reben/emeten, el bus que fan servir, etc.

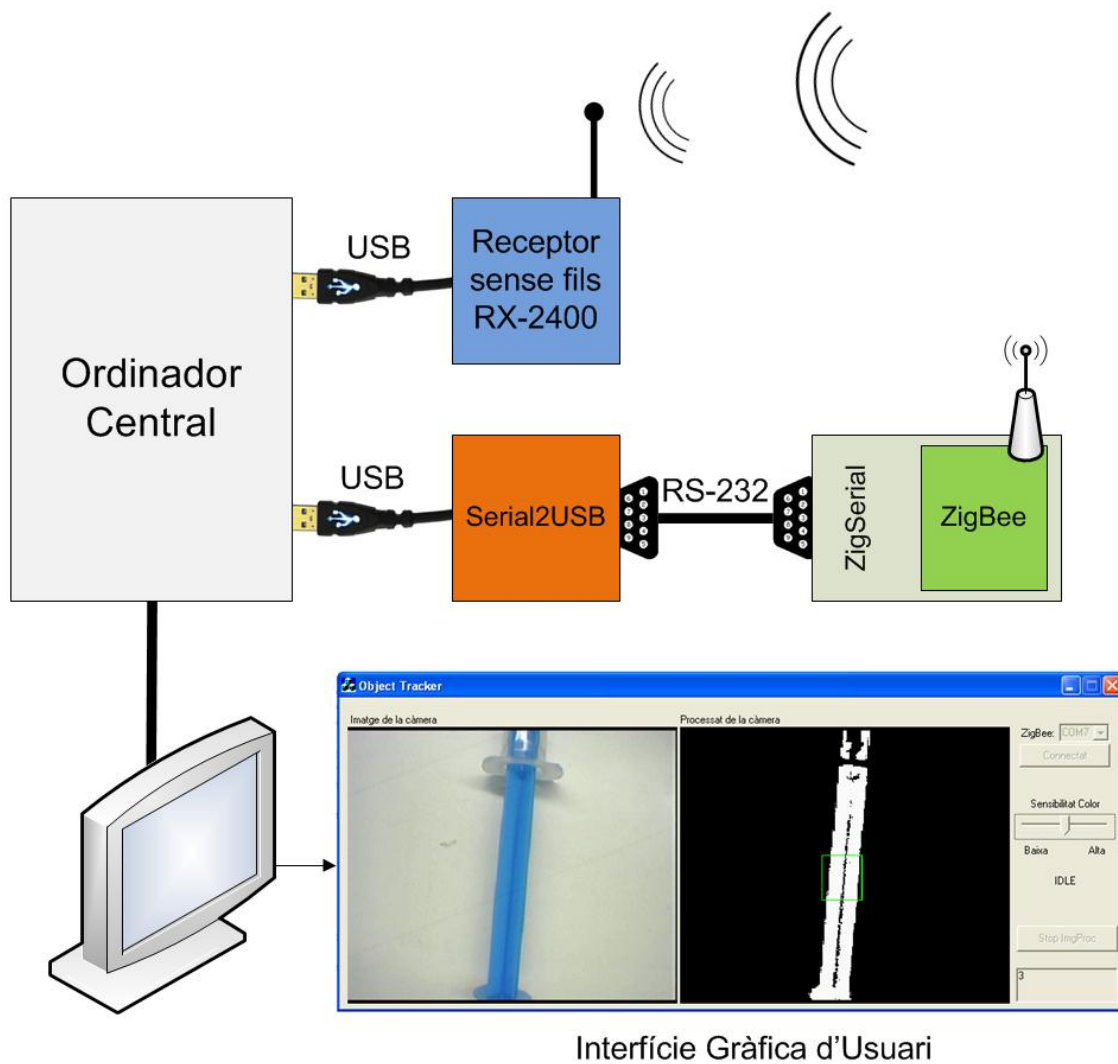


Figura 17 – Esquema de connexions dels diferents components a l'ordinador central.

#### **6.2.2.1. Receptor sense fils RX-2400**

El receptor sense fils RX-2400 és un component del kit de desenvolupament de Bioloid que es connecta a l'ordinador a través de la interfície USB. L'objectiu d'aquest component és la recepció de les imatges provinents de la càmera sense fils instal·lada a un dels braços robòtics.

Permet ser configurat per rebre dades entre quatre canals de comunicació sense fils diferents. En el nostre cas particular, tant la càmera com el receptor estan configurats per treballar pel canal 3<sup>18</sup>. **L'ús d'aquest component, juntament amb la càmera sense fils, lliga el funcionament de la plataforma robòtica a un ordinador.**

#### **6.2.2.2. Zig2Serial i Serial2USB**

Per poder rebre i transmetre dades amb el mòdul ZigBee de l'ordinador central, i en conseqüència, poder interactuar amb els robots del sistema, calen dos components addicionals, també inclosos en el kit de desenvolupament de Bioloid.

El primer d'aquests components s'anomena Zig2Serial, i es tracta d'una placa electrònica a la qual s'hi afegeix el mòdul ZigBee (aplicant unes soldadures) i amb els circuits electrònics associats permet comunicar-se a través del port sèrie (protocol RS232) de l'ordinador.

El segon component addicional, tot i que no estrictament necessari, és un convertidor de comunicacions RS232 a USB, que permet fer servir aquest protocol de comunicacions més modern com si es tractés d'un port sèrie més del sistema.

La combinació d'aquests dos elements addicionals permet fer servir el mòdul ZigBee senzillament connectant-lo a un cable USB a l'ordinador central.

---

<sup>18</sup> L'ús d'un canal o un altre no afecta al rendiment de l'aplicació. Els canals són subdivisions de la banda de freqüència assignada.



A la figura 17 es pot observar la connexió d'aquests dos components, i a les figures 18 i 19 es poden observar els components fotografiats.

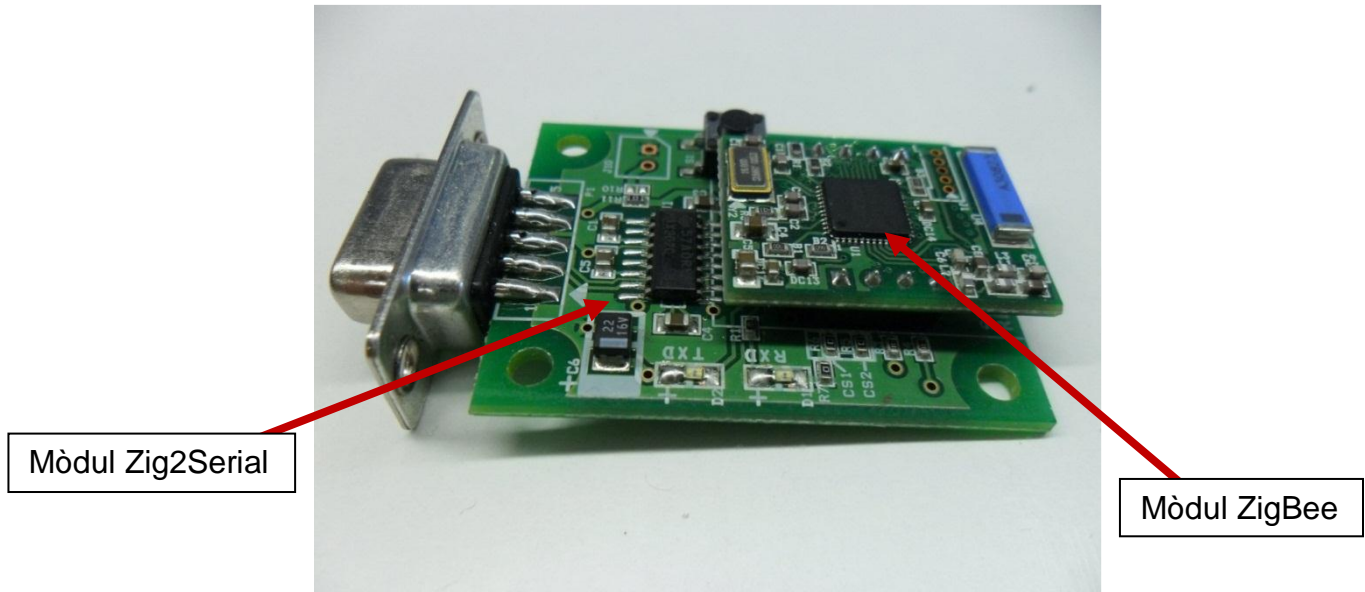


Figura 18 – Mòdul Zig2Serial amb el mòdul ZigBee connectat.

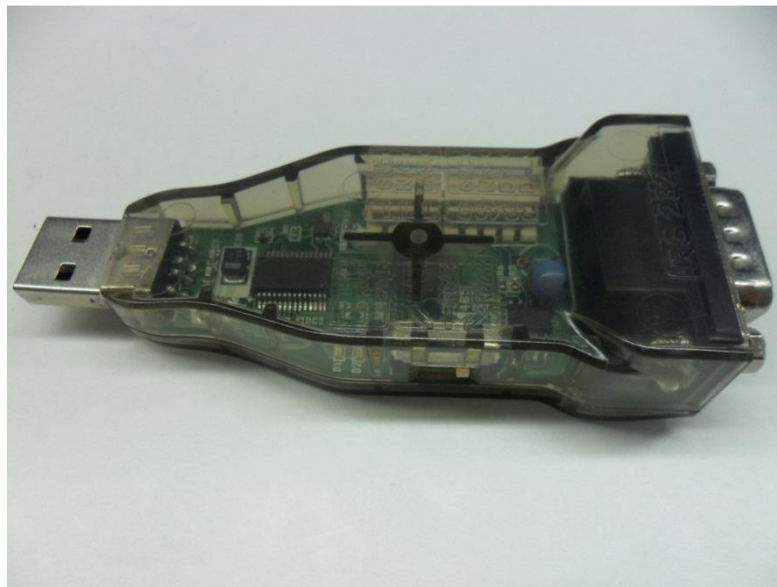


Figura 19 – Mòdul Serial2USB.

Veiem que el Serial2USB permet connectar el Zig2Serial pel connector de la dreta (RS232) i connectar-se a l'ordinador pel port USB de l'esquerra.



Amb aquests darrers elements es dona per finalitzat l'apartat d'introducció als components físics integrants del projecte, on s'ha pogut analitzar l'arquitectura de tot el sistema, la morfologia dels robots, el hardware necessari per dur a terme la comunicació entre els mòduls del sistema, etc.

A l'apartat següent es cobrirà tot el que fa referència a la programació d'aquest sistema, concretament el que fa referència al codi dels robots –firmware-.

## 7. Programació del robot. El firmware.

Fins ara aquest projecte ha tocat temes tan diversos com l'electrònica, la manipulació d'eines i material industrial, etc. Això és així degut a que els projectes de robòtica sovint van acompanyats de tasques de preparació de l'entorn, muntatge del sistema i realització de jocs de proves o demostracions en entorns realistes.

Per a assolir els objectius funcionals que ens hem proposat -bàsicament la manipulació d'instrumental quirúrgic abans i durant una cirurgia- cal programar tota una lògica de control per donar la intel·ligència necessària a un sistema construït amb elements hardware de baix nivell, i així poder complir els objectius marcats.

En aquest apartat es detallaran els punts més destacables de la implementació del codi, es posaran de manifest les decisions preses, la seva motivació i, si s'escau, possibles alternatives.

A partir d'ara, doncs, la memòria es centrarà en la part purament informàtica del projecte, descrivint tots i cada un dels punts concrets que aporten intel·ligència a la plataforma robòtica que s'ha dissenyat.

### **7.1. Abstracció del hardware cap al software. Del WriteByte al setRobot.**

Aquest apartat és un dels pilars sobre els quals s'ha basat el projecte. Es tracta d'analitzar i desenvolupar una solució que ens permeti pujar diversos esglaons en l'abstracció, sent capaços de programar el robot parlant d'articulacions i de posicions, en comptes de motors, bytes i adreces.

Per visualitzar l'objectiu d'aquest sistema, a la figura 20 es poden observar els diferents nivells d'abstracció que s'han seguit.

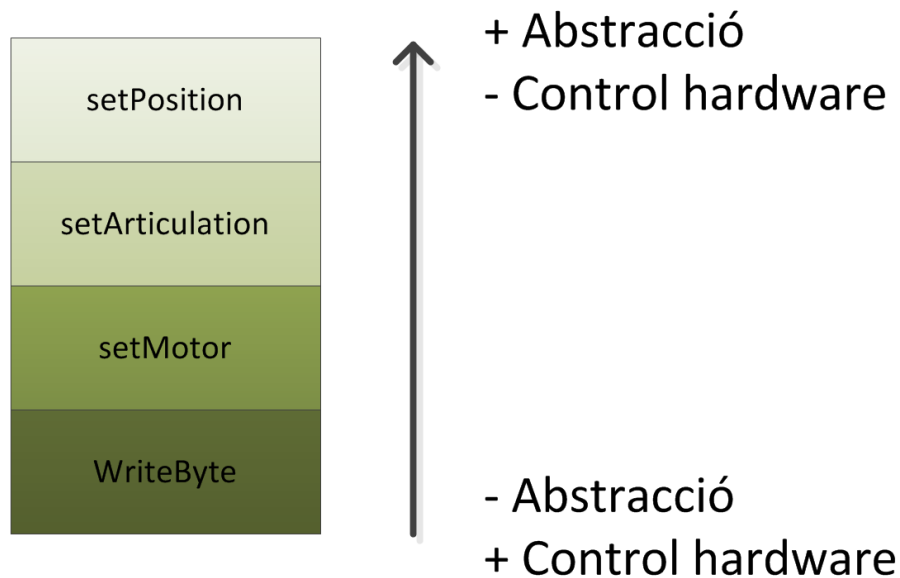


Figura 20 – Nivells d'abstracció del codi.

A la descripció del sistema ja s'ha parlat de com està construït el robot; bàsicament són servomotors connectats a través d'un bus sèrie de dades. Tot seguit es farà una explicació de la programació a més baix nivell, sobre la qual es basa tota la resta de codi. És important saber quin és el nivell de detall que ens permet l'API del fabricant, per tal de comprendre algunes de les limitacions que s'han tingut a l'hora de desenvolupar les instruccions d'alt nivell.

Connectats al bus sèrie, els servomotors tenen associat un identificador propi, és a dir una adreça, que els permet saber si una comanda de lectura o escriptura de registres està dirigida a ell o no. Així doncs, el més bàsic que es permet fer és la lectura i escriptura dels bytes d'aquests registres, amb les següents instruccions.

```
byte ReadByte(MOTOR_ID, ADDRESS);
```

```
void WriteByte(MOTOR_ID, ADDRESS, VALUE);
```

És amb aquestes dues instruccions, doncs, amb les que ens basarem el que queda de projecte. Tot seguit es mostra una taula amb les adreces que s'han fet servir.

Taula 1 – Registres dels servomotors utilitzats en aquest projecte.

ADDRESS	Contingut
P_GOAL_POSITION_L	Bytes de menys i més pes de l'adreça que indica al servomotor la posició destí.
P_GOAL_POSITION_H	
P_GOAL_SPEED_L	Bytes de menys i més pes de l'adreça que indica al servomotor la velocitat a la qual s'ha de desplaçar.
P_GOAL_SPEED_H	
P_CURRENT_POSITION_L	Bytes de menys i més pes de l'adreça per llegir la posició actual del servomotor.
P_CURRENT_POSITION_H	
P_CURRENT_TORQUE_L	Bytes de menys i més pes de l'adreça per llegir el parell motor actual del servomotor.
P_CURRENT_TORQUE_H	

A partir d'aquí, tot el que es vagi construïnt per sobre ens permetrà abstroure'ns cada cop més, i oblidar el concepte de MOTOR\_ID i pensar cada cop més en articulacions i posicions.

Primer de tot cal definir com interactuarem amb els motors. Degut a que la posició desitjada es defineix amb un valor enter de 10 bits (de 0 a 1023), és necessari escriure al byte de més pes (HIGH) i al de menys pes (LOW) del registre-adreça associat a la posició del motor. Aquesta tasca és repetitiva i la causa d'un codi poc elegant i poc comprensible, per tant el primer que s'ha fet ha estat pujar un nivell d'abstracció i definir la funció:

```
void setMotor(MOTOR_ID, POSITION, SPEED);
```

L'ús d'aquesta rutina és intuïtiu i permet simplificar el codi. Tot seguit un exemple d'implementació, on s'observa la partició dels paràmetres POSITION i SPEED en dos bytes, i la seva posterior transmissió al servomotor determinat.

```
375 void setMotor(byte mid, word pos, word vel ){  
376     byte pos_H;  
377     byte vel_H;  
378  
379     vel_H=vel>>8;  
380     WriteWord(mid, P_GOAL_SPEED_L, vel) ;  
381     WriteWord(mid, P_GOAL_SPEED_H, vel_H) ;  
382  
383     pos_H=pos>>8;  
384     WriteWord(mid, P_GOAL_POSITION_L, pos) ;  
385     WriteWord(mid, P_GOAL_POSITION_H, pos_H) ;  
386 }
```

Figura 21 – Implementació de la rutina setMotor.

Avançant en l'abstracció, com s'ha comentat anteriorment el robot està format per 6 articulacions i l'eina; cada articulació, però, està formada per un motor diferent, havent-hi inclús articulacions que la conformen dos motors diferents. Aquest fet, sumat al de l'existència de dos robots, fa molt difícil la programació del sistema només basant-nos en els identificadors dels motors, i ens obliga a anar més enllà, i pujar un nivell més de la pila d'abstracció: les **articulacions**.

Definint un valor constant per a cada articulació –independentment del robot que estigui executant el codi- i gestionant adequadament quin robot està executant el codi i quins motors conformen cada articulació, podem definir el joc d'instruccions següent:

```
void setArticulation(ART_ID, POSITION, SPEED, BLOCKING);  
  
word getArticulation(ART_ID);
```

Tot seguit es presenta una implementació de la funció setArticulation.

```

375 void setArticulation(byte art, word pos, word vel, byte blocking){
376     short inPos=0;
377     short pAct=0;
378
379     if(pos>1023) pos=1023;
380     if(pos<0) pos=0;
381
382     switch(art){
383         //Excepció per articulacions amb 2 motors
384         case ESPATLLA:
385             if(robCon==ID_ROBCAM){
386                 setMotor(2, pos, vel);
387                 setMotor(3, pos, vel);
388             }else if(robCon==ID_ROBIR){
389                 setMotor(10, pos, vel);
390                 setMotor(11, pos, vel);
391             }
392             break;
393         //Excepció per articulacions amb 2 motors
394         case COLZE:
395             if(robCon==ID_ROBCAM){
396                 setMotor(4, pos, vel);
397                 setMotor(8, pos, vel);
398             }else if(robCon==ID_ROBIR){
399                 setMotor(12, pos, vel);
400                 setMotor(13, pos, vel);
401             }
402             break;
403         //Cas general. 1 articulació -> 1 motor
404         default:
405
406             setMotor(motorIDs[art], pos, vel);
407             break;
408     }
409
410     //Gestió del bloqueig de posicionament
411     if(blocking){
412         while(!inPos){
413             pAct=getArticulation(art);
414             if(abs(pos-pAct) < THLD_BLOCK){
415                 inPos=1;
416             }
417         }
418     }
419 }
    
```

Figura 22 – Implementació de la rutina setArticulation.

Com es pot observar, només en els casos excepcionals de 2 motors per articulació hem de programar manualment l'identificador dels motors en qüestió. Per tots els altres casos, la gestió de quin identificador de motor es fa

servir és totalment transparent, associada al vector d'identificadors de motors que es descriurà en aquest mateix apartat.

Tot seguit farem el darrer dels passos en l'abstracció del sistema de baix nivell fins a la programació del robot en alt nivell. Un cop es controlen les articulacions, el darrer pas és tenir el control sobre la totalitat de les articulacions del robot, és a dir, sobre la seva posició global. Això s'aconsegueix amb la següent rutina:

```
void setRobot(POSITION, SPEED, BLOCKING)
```

En aquest cas, la posició d'un robot sencer ve definida per un vector de posicions de cada un dels seus motors. D'aquesta manera, amb un joc de posicions fixes, es poden dur a terme les tasques necessàries de forma senzilla, sacrificant un punt de versatilitat.

```
word staticPosition[POSICIONS][MOTORS] = {  
    {506,660,660,140,140,520,464,519,512}, //1.POS_INIT  
    {700,488,483,202,200,520,275,348,510}, //2.POS_OBSERVA  
    {750,392,392,215,215,524,361,422,511}, //3.POS_RECULL
```

Figura 23 – Fragment de posicions globals dels robots predefinides al sistema.

Com s'observa, igual que el posicionament de les articulacions, aquesta instrucció també permet el bloqueig del codi fins que el robot no estigui completament posicionat.

La implementació d'aquesta rutina es pot veure tot seguit:

```

435 void setRobot(byte pos, word vel, byte blocking){
436     short inPos=0;
437     short pAct=0;
438
439     setArticulation(CANELL2 ,staticPosition[pos][CANELL2] ,vel,0);
440     setArticulation(COLZE ,staticPosition[pos][COLZE] ,vel,0);
441     setArticulation(ESPATLLA,staticPosition[pos][ESPATLLA],vel,0);
442     setArticulation(BASE ,staticPosition[pos][BASE] ,VEL_MED,0);
443     setArticulation(CANELL1 ,staticPosition[pos][CANELL1] ,vel,0);
444     setArticulation(CANELL3 ,staticPosition[pos][CANELL3] ,vel,0);
445
446     if(blocking){
447         while(!inPos){
448             pAct = getArticulation(BASE);
449             if(abs(staticPosition[pos][BASE]-pAct) < THLD_BLOCK){
450                 pAct = getArticulation(ESPATLLA);
451                 if(abs(staticPosition[pos][ESPATLLA]-pAct) < THLD_BLOCK){
452                     pAct = getArticulation(COLZE);
453                     if(abs(staticPosition[pos][COLZE]-pAct) < THLD_BLOCK){
454                         pAct = getArticulation(CANELL1);
455                         if(abs(staticPosition[pos][CANELL1]-pAct) < THLD_BLOCK){
456                             pAct = getArticulation(CANELL2);
457                             if(abs(staticPosition[pos][CANELL2]-pAct) < THLD_BLOCK){
458                                 pAct = getArticulation(CANELL3);
459                                 if(abs(staticPosition[pos][CANELL3]-pAct) < THLD_BLOCK){
460                                     inPos=1;
461                                 }
462                             }
463                         }
464                     }
465                 }
466             }
467         }
468     }
469 }
    
```

Figura 24 – Implementació de la rutina setRobot.

Consideracions sobre aquesta implementació.

1. De les tres velocitats que s'han definit pel moviment de les articulacions del robot, la base sempre es posiciona a velocitat mitja. Les possibles friccions entre la base del robot i l'eix del servomotor quan aquest gira a certa velocitat, obren la possibilitat a un error en el posicionament global del robot. S'ha decidit doncs que minimitzant la velocitat , i per tant el parell del motor, de la base s'intenta minimitzar el risc d'error en el posicionament.



2. Cal diferenciar entre posicionar el robot i posicionar les articulacions de forma bloquejant. Si caiguéssim en l'error d'implementar el bloqueig del posicionament del robot fent que cada moviment d'articulació fos bloquejant, ens trobaríem amb la situació que el robot s'aniria movent articulació a articulació. Com no és aquest el nostre objectiu, es transmeten les comandes de posicionament a cada articulació, i després es gestiona el bloqueig.

## 7.2 Instruccions de posicionament bloquejants

Com s'ha començat a veure a l'apartat anterior, una característica de programació ha estat la necessitat de programar un sistema de bloqueig de les instruccions de posicionament amb la finalitat de poder marcar un determinat ritme a l'execució d'aquestes instruccions, i per tant al moviment final del robot.

La motivació principal d'aquesta implementació ha estat la forma de treballar que tenen els servomotors Dynamixel. Quan hi ha una llista d'instruccions de posicionament, la controladora les executa tot enviant la comanda concreta al servomotor indicat, i tot seguit passa a la següent instrucció. D'aquesta manera no es pot saber si el motor ha assolit la nova posició, ni quan ho ha fet, ni si ha tingut algun problema durant el trajecte.

Malauradament la solució trobada passa per la tècnica del *polling*<sup>19</sup>, preguntant constantment al motor si ha assolit la nova posició. La solució implementada permet posicionar articulacions i el robot sencer fent ús del *polling* o no, en cas de voler mantenir un ordre determinat d'execució o no. Seguidament es mostra una figura aclaridora del codi:

```
1 setRobot(POS_INIT_G, VEL_MED,1);  
2 setArticulation(TOOL, OBERTA, vel, 0);  
3 setRobot(POS_OBSERVA, VEL_MED,1);
```

Figura 25 – Exemple d'instruccions bloquejants.

<sup>19</sup> [http://en.wikipedia.org/wiki/Polling\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Polling_(computer_science))

Posant atenció al tercer paràmetre de les instruccions, en aquest cas primer es posicionaria el robot a la posició inicial i el programa esperaria a que aquest hi arribés. Tot seguit executaria la instrucció 2, obrint la pinça del robot –és a dir, enviaria la comanda al servomotor concret- però no esperaria a que s'obris, tot executant la següent instrucció de posicionament (3), movent el robot a la posició d'observació.

A les figures 22 i 24 hi ha la implementació de les rutines de posicionament on es pot observar també la implementació del sistema de bloqueig per *polling*.

### **7.3 Comunicació bidireccional sense fils entre els robots i el PC de suport**

En aquest apartat s'explicarà pas a pas com s'ha implementat la solució trobada al problema derivat de la comunicació necessària entre els mòduls del sistema.

Es farà una explicació prèvia de les aproximacions prèvies que s'han considerat i es raonarà el motiu pel qual s'han deixat de banda, tot argumentant perquè es creu que la solució final proposada és la més encertada.

A la figura següent es mostra un diagrama simplificat amb la connectivitat dels diferents sistemes de la plataforma.

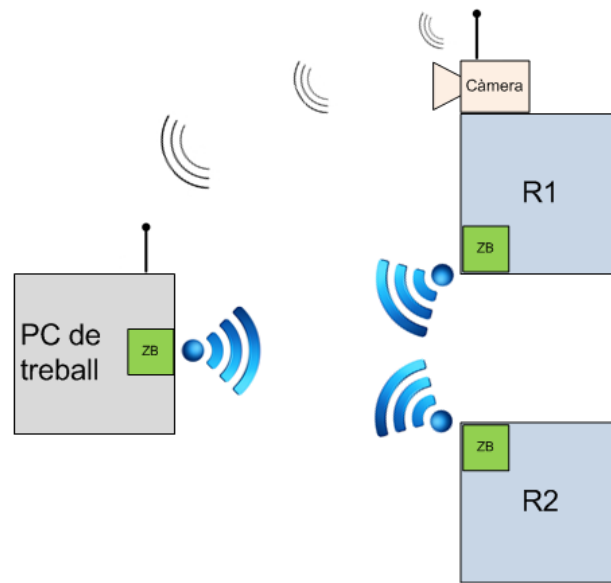


Figura 26 – Esquema de comunicació simplificat.

Com s'ha comentat en apartats anteriors –i s'observa a la figura anterior-, el problema de dotar de connectivitat sense fils als actors del sistema ha estat resolt incorporant mòduls ZigBee als dos robots i a l'ordinador central.

A continuació observem el mateix diagrama, però amb les dades que han de fluir entre els mòduls del sistema.

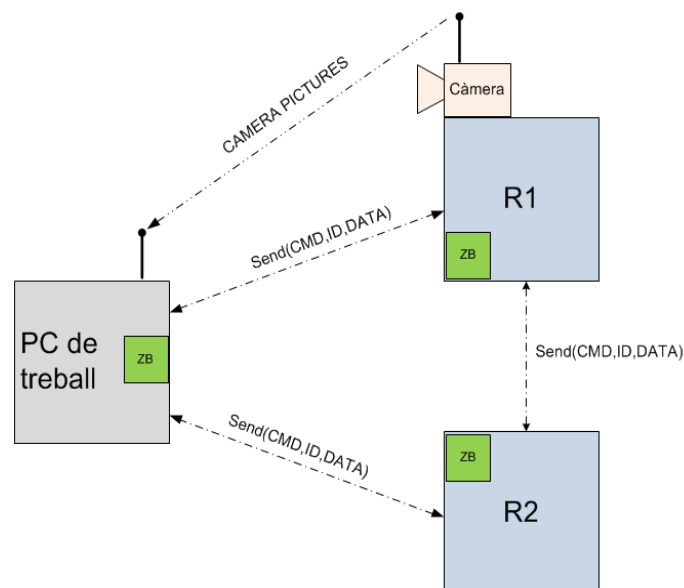


Figura 27 – Esquema amb les dades del sistema de comunicació.

Gràcies a les llibreries de programació d'aquests mòduls ZigBee, proporcionades pel fabricant, es disposa d'un joc d'instruccions senzill que permet configurar el mòdul i fer-lo servir de la següent manera:

- Configurar l'adreça de destinació dels paquets de dades. A part de l'adreça pròpia de cada mòdul, aquest pot ser configurat amb l'adreça pròpia del mòdul destinatari del paquet de dades. Aquesta adreça és un valor de 16 bits que, configurat a 0xFFFF, el mòdul ZigBee interpretarà que es troba en mode *broadcasting*, enviant informació a múltiples receptors.
- Rebre dades. La recepció es fa a través un *flag*<sup>20</sup> que ens indica si el mòdul ha rebut noves dades, i d'una variable de 16 bits que emmagatzema el valor del paquet rebut. Tot aquest sistema és transparent a l'usuari.
- Transmetre un paquet a una adreça destí. El paquet el conforma una paraula de 16 bits.

El joc d'instruccions concret es pot observar a la figura següent.

```
120 extern byte ZigbeeSet(byte Function, word newDest, byte newWait);  
121 extern void ZigbeePacketCheck(void);  
122 extern void ZigbeePacketTransfer(word wData);
```

Figura 28 – Joc d'instruccions ZigBee de la llibreria de programació del fabricant.

<sup>20</sup> Flag: variable típicament booleana que indica la ocurrència d'un esdeveniment determinat.

El paràmetre **Function** de la funció ZigbeeSet pot tenir els valors següents:

Taula 2 – Valors del paràmetre **Function** de la funció ZigbeeSet.

Valor del paràmetre <b>Function</b>	Descripció
0	Llegeix i actualitza els valors emmagatzemats al mòdul. Ens permet obtenir l'adreça física del propi mòdul i l'adreça destí que té configurada.
1	Permet configurar l'adreça destí amb el paràmetre <b>newDest</b> .
2	Permet configurar el 'WaitMode' (mode d'espera) del mòdul <sup>21</sup> .
3	Permet configurar l'adreça destí, escrita al paràmetre <b>newDest</b> i el 'WaitMode', al paràmetre <b>newWait</b> .

Així doncs, amb el sistema llest per a comunicacions sense fils, passem tot seguit a quins objectius de comunicació es volien assolir.

### 7.3.1 Objectius de la implementació del sistema de comunicació

Els objectius principals de la implementació de tot el sistema de comunicació sense fils dels robots, són:

1. Enviar un màxim de 16 comandes diferents a qualsevol dels altres sistemes de la plataforma. Cada comanda ha de portar una trama de dades associada que permeti transmetre la informació necessària.
2. Ser eficient amb la manera com es transmet la informació. El fet de transmetre un valor representable, p.ex., amb 5 bits utilitzant 16 bits d'informació és un motiu d'ineficiència.

<sup>21</sup> A la documentació facilitada per Bioloid no apareix cap explicació del paràmetre newWait, i donat que als fragments d'exemple que proporcionen no en fan ús, assumim que no interfereix en l'ús normal del mòdul.

3. Independència de l'ordinador central. Dotar al sistema de la capacitat de comunicar-se robot a robot sense la necessitat de passar per l'ordinador principal, en cas de ser necessari.
4. Comunicar de forma fiable tots els actors del sistema. Que les dades arribin al seu destinatari.

### **7.3.2 Implementació de les comunicacions sense fils**

Donat que els mòdul ZigBee tenen un sistema d'adreçament propi i permeten la transmissió de paraules de dades de 16 bits, això obre la porta a moltes possibles implementacions que satisfacin els objectius que s'han descrit anteriorment.

Per aquesta raó s'ha decidit provar diferents aproximacions sobre com implementar la comunicació sense fils des dels robots. A continuació hi ha una descripció de les 4 aproximacions que s'han considerat, tenint en compte els objectius descrits anteriorment.

#### **7.3.2.1 Aproximació #1 - Configurar el mòdul ZigBee manualment abans d'enviar cada paquet**

Aquesta primera aproximació vol aprofitar els 16 bits de dades que es permet enviar a cada paquet, fent ús del sistema d'adreçament propi dels mòduls ZigBee.

Una possible solució passa per enviar dos paquets d'informació, un amb la comanda i l'altre amb les dades. Aquesta solució queda descartada directament pel fet de d'utilitzar 16 bits de dades -és a dir, un valor fins a 65.353-, per a codificar les 16 comandes; això va contra l'objectiu de fer eficients les comunicacions.

Així doncs, l'alternativa passa per utilitzar els 4 bits de més pes com a identificador de la comanda i els 12 bits restants com a dades. La trama de dades queda definida a la figura 29.

Per més comoditat durant aquest apartat, es farà servir la següent llegenda:

- C – bit de comanda
- D – bit de dades.
- A – bit d'adreça.
- N – bit de nombre de bytes que s'enviaran.

C	C	C	C	D	D	D	D	D	D	D	D	D	D	D	D
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figura 29 – Ús del paquet de dades en l'aproximació #1.

### 7.3.2.2 Aproximació #2 - ZigBee configurat en broadcast. Un paquet de capçalera i un paquet de dades.

Aquesta aproximació elimina la pèrdua de bits útils a la paraula de dades, ja que en aquest cas s'envien dos paquets per cada transmissió: un amb la informació del destinatari i la comanda que es vol transmetre -la capçalera-, i l'altre amb les dades desitjades. En aquest mode és necessari implementar un lògica que identifiqui els paquets rebuts i, a més a més, un sistema per tal de que el robot tingui constància de que, si l'identificador al primer paquet es correspon amb el seu propi identificador, el següent paquet és per ell. Això genera codi més complex i, com veurem en el test de rendiment, no és gens eficient.

Paquets de dades:

C	C	C	C	C	C	C	C	A	A	A	A	A	A	A	A
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figura 30 – Primer paquet de dades de l'aproximació #2.

D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figura 31 – Segon paquet de dades de l'aproximació #2.

### 7.3.2.3 Aproximació #3 - ZigBee configurat en broadcast. Un paquet de capçalera i N paquets de dades.

Aquesta aproximació passa per incloure a la capçalera el nombre de paquets de dades que s'enviaran posteriorment. Aquesta implementació cerca més eficiència a l'hora de transmetre dades.

La implementació hauria de controlar el paràmetre concret de paquets que es transmetran, a més a més de gestionar correctament el nombre de paquets que falten per llegir, si es tracta del robot receptor, o el número de paquets que s'han de rebutjar, si tracta d'un actor del sistema que no sigui el destinatari de la comunicació.

Ús dels paquets de dades:

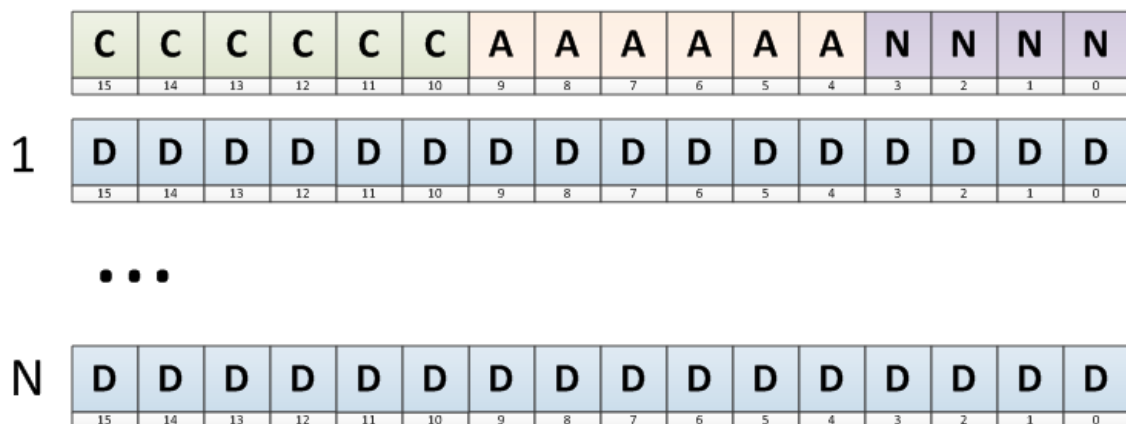


Figura 32 – Paquets de dades de la implementació #3.



S'observa que a la capçalera es fan servir 6 bits per codificar la comanda, 6 més per codificar l'adreça destí, i 4 bits per codificar el nombre de paquets que es transmetran a continuació.

#### 7.3.2.4 Aproximació #4 - ZigBee configurat en broadcast. Un paquet.

Aquesta aproximació cerca integrar en un sol paquet de dades la comanda que s'envia, juntament amb l'identificador del destinatari i les dades associades. D'aquesta manera s'elimina la problemàtica d'haver d'enviar dos paquets de dades o d'haver de reconfigurar el mòdul ZigBee cada cop que es vol utilitzar.

La trama de dades es proposa així.

C	C	C	C	A	A	D	D	D	D	D	D	D	D	D	D
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figura 33 – Paquet de dades de l'aproximació #4.

Aquesta aproximació requereix igualment d'una petita lògica de control al firmware del robot i al codi de l'aplicació per a PC. Aquesta lògica s'encarregarà de descodificar el paquet rebut i d'identificar els tres camps que s'hi troben.

#### 7.3.3 Proves de rendiment

Per poder avaluar quin sistema era el més adequat, s'han dut a terme unes proves de rendiment per a cada una de les aproximacions.

La prova de rendiment s'ha fet enviant un total de 16.000 Bytes de dades útils – *payload*- d'una controladora a l'altra –no confondre amb 16.000 Bytes de dades, capçaleres incloses-. Com no totes les aproximacions tenen la mateixa

capacitat de transportar dades útils –columna ‘Payload (%)’-, s’ha hagut d’ajustar el valor dels paquets enviats, i precisament això ha originat les discrepàncies de temps entre una aproximació i una altra.

Tot seguit es pot veure una taula amb les dades del test de rendiment que s’ha dut a terme en les 4 possibles aproximacions a solucionar les comunicacions sense fils del sistema.

Taula 3 – Taula amb els resultats de les proves de rendiment *WiFi*.

<b>Test de rendiment – Payload total 16.000 Bytes</b>				
Aproximació	Payload (%)	# paquets enviats	Temps (s)	Ample de banda (KB/s)
#1	75	10.667	11,04	1,42
#2	50	16.000	17,23	0,91
#3 (N=8)	89	9.000	9,55	1,64
#4	62,5	12.800	13,12	1,19

La primera observació que es fa és que hi ha una relació clara entre el nombre de paquets enviats i el temps que es triga, que ens permet deduir un ample de banda real aproximat de 2 KB/s.

La segona observació és que l’aproximació optimitzada per a un gran volum de dades, la #3, és la que millor rendiment té, fet que no sorprèn, tenint en compte l’optimització aplicada.

Finalment cal comentar que ens vam trobar amb la restricció de no poder fer servir l’aproximació #1, ja que en el moment d’utilitzar les llibreries de programació pel ZigBee connectat a l’ordinador, es va comprovar que no permetia canviar l’adreça de destí en temps d’execució.

### 7.3.4 Conclusions

La conclusió a la que s'ha arribat, degut a la restricció comentada anteriorment, i a que les necessitats del sistema no requereixen transmetre un gran volum de dades, és que l'aproximació més adequada és la #4, per ser la que ofereix millor rendiment després de la #1, i pot ser implementada més fàcilment que la #3.

Tot seguit es mostra una possible implementació d'aquesta aproximació, situada al bucle principal del codi dels dos robots. Bàsicament comprova l'existència d'un nou paquet de comunicacions, el descodifica i crida a la rutina d'atenció a les comunicacions sense fils, que s'encarrega de gestionar la comanda rebuda i les dades associades.

```
while(1){
    cicles++;
    ZigbeePacketCheck();
    if(gbNewPacket){ //flag indicador de nou paquet.
        gbNewPacket=0;
        dades=gwZigbeeRxData;
        //identificador del robot (o PC) destí
        robID=(dades&0x0C00);
        //comanda rebuda
        cmd =(dades&0xF000);
        //dades (payload) del paquet
        dades=(dades&0x03FF);

        if(verbose) printf("MSG_WIFIRCVD");
        //Si el paquet és per nosaltres o es un paquet broadcast a 00H
        if(robID==robCon || robID==ID_BROADCAST){
            //Rutina de Servei a la Comunicació Wifi (cmd,dades)
            _RS_WifiCom();
        }
    }
}
```

Figura 34 – Implementació de l'aproximació #4.

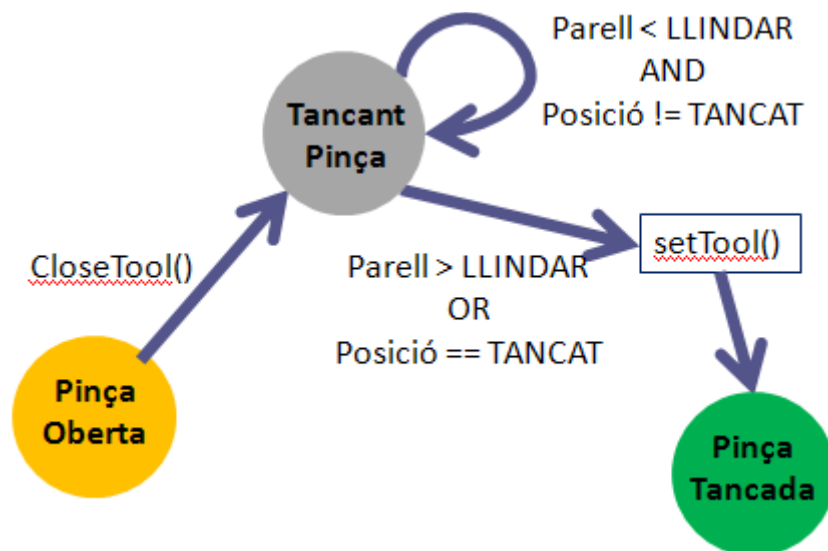
#### **7.4 Control de pressió durant el tancament de la pinça del robot**

Una funció afegida al sistema ha estat la implementació del control de pressió de l'eina del robot. Aquesta tasca ha estat necessària degut a les limitacions tècniques dels servomotors disponibles.

El context és el següent: Als motors que utilitzem se'ls hi pot indicar la posició a la qual han d'anar, especificant també la velocitat desitjada. Això ens pot fer pensar en definir dues posicions, una per tenir la pinça oberta i una altra per tenir la pinça tancada. Aquesta primera aproximació produeix un error immediatament després d'intentar agafar un objecte d'un gruix considerable. L'explicació és senzilla; el servomotor té la consigna d'arribar a la posició de tancat, i quan hi posem un objecte al mig, aquesta consigna de posició no es veu satisfeta i després d'uns pocs segons fent pressió, el motor entra en estat de fallada.

La solució a la qual s'ha arribat és la que té en compte el parell motor que exerceix el servomotor en tot moment. Els motors d'aquest kit de desenvolupament permeten consultar el parell que estan exercint i, juntament amb la informació sobre la posició actual del motors, podem dur a terme un bucle de consulta del qual se'n surti si el parell exercit és major a un llindar determinat o bé si s'ha arribat a la posició desitjada. Així doncs, es donarà per tancada la pinça si arriba a la seva posició natural de tancament, o si troba algun objecte al mig.

A continuació es presenta una màquina d'estats representant aquesta funcionalitat:

Figura 35 – Màquina d'estats de la funció `CloseTool()`.

Finalment cal comentar un detall força important d'aquesta implementació. Es tracta de la instrucció final del tancament de la pinça. Aquesta instrucció ha de re-configurar el servomotor per tal que tingui com a posició desitjada la última posició amb la qual s'ha sortit del bucle de control. Això és important pel que s'ha explicat abans, ja que de no fer-ho, per molt que el sistema detecti que hi ha un objecte que impedeix arribar a la posició de tancament, el servomotor seguirà aplicant el mateix parell i acabarà entrant en mode de fallada.

### 7.5 Unificació del codi. Un codi per $N$ robots.

Aquest punt tracta sobre la decisió que es va prendre al principi del projecte de programar un únic codi per a tots dos robots. Les motivacions d'aquesta decisió són les següents:

- Facilitat a l'hora de desenvolupar. No cal compilar dues vegades i mantenir actualitzades les dues versions del codi, una per cada robot.
- Parametrització del sistema. Com s'explicarà més endavant, aquesta decisió permet connectar al sistema el número de robots que es desitgi, essent el firmware l'encarregat de detectar quin robot està gestionant i

parametritzar-se automàticament. Aquest fet dóna una transparència enorme en quant a programació i escalabilitat del sistema.

- Quantitat de memòria suficient. Pel sistema dissenyat en aquest projecte, la memòria del microcontrolador no ha estat un factor a tenir en compte. Així com sempre s'ha mantingut una política estricta de utilitzar el tipus de variables adequades, a l'hora de poder utilitzar el mateix codi en diversos robots, s'ha permès la declaració de variables que només seran utilitzades en un braç robòtic.

Sabem que els robots d'aquest sistema són exactament iguals; tenen la mateixa controladora i en la majoria dels aspectes es comporten igual. La comunicació sense fils es du a terme en mode *broadcast*, com hem vist a l'apartat anterior, tenen el mateix nombre d'articulacions, la mateixa eina, la comunicació de la càmera es fa de forma transparent al robot i, el que és més important, perseguim l'objectiu de desenvolupar un sistema bi-robòtic el més transparent possible cap a l'usuari final. Així doncs, sembla evident que la programació d'un mateix codi parametritzat per ser executat en dos robots diferents és l'aproximació ideal.

Les característiques principals es troben descrites en el següent punt.

### **7.5.1 Inicialització del sistema**

Quan iniciem el sistema –movent el switch 'ON-OFF' a la posició ON- la controladora executa la subrutina `_Initialize()`, que gestiona la inicialització de les entrades i sortides del microcontrolador, dels temporitzadors encarregats de la comunicació sèrie, la gestió de la càrrega de la bateria, la configuració del mòdul ZigBee, etc. Tot just després, un cop el sistema està a punt, s'executa la subrutina anomenada `DetectaRobot()`, que retorna l'identificador del robot connectat.

```
6      _Initialize();
7
8      if(DetectaRobot()!=0){
9          if(verbose) printMessage(MSG_INITROB);
10
11         setRobot(POS_OBSERVA, VEL_MED,1);
12         if(robCon==ID_ROBIR)    setRobot(POS_INIT_G,    VEL_MED,1);
13         if(robCon==ID_ROBCAM)  setRobot(POS_INIT,      VEL_MED,1);
14     }
```

Figura 36 – Primeres instruccions del firmware del robot.

A continuació observarem més detingudament la implementació de la subrutina que identifica el robot que hi ha al sistema.

```
555 word DetectaRobot(void) {
556     if( (ReadWord(9, P_ID)) == 0x0109){
557         robCon=ID_ROBCAM;
558         motorIDs=&motors[0];
559         printMessage(MSG_ROBCAM);
560     }else if( (ReadWord(1, P_ID)) == 0x0101){
561         robCon=ID_ROBIR;
562         motorIDs=&motors[1];
563         printMessage(MSG_ROBIR);
564     }else{
565         robCon=-1;
566         printMessage(MSG_NOROB);
567     }
568     return robCon;
569 }
```

Figura 37 – Implementació de la rutina DetectaRobot.

Tot seguit s'analitzen els punts més importants d'aquesta rutina.

En primera instància s'observa com, irremeiablement, apareixen identificadors i adreces físiques dels servomotors. Això és així ja que per saber quin robot està connectat, el sistema ha de tenir alguna informació característica per identificar-los, i en el cas actual, es tracta de l'identificador del motor que està situat a la base del robot. Així doncs, en aquesta subrutina hi haurà tants else if com

nombre de robots es vulgui connectar, i cada un comprovarà que l'identificador de la base sigui el que té parametritzat.

Si es produeix alguna coincidència –és a dir, el sistema reconeix el robot que té connectat- es produeix la parametrització del sistema. Aquesta parametrització, en el sistema actual és molt bàsica, però queda palès que es podria estendre tant com el programador volgués.

Concretament s'assigna a la variable `motorIDs` la posició de l'estructura de dades que conté la distribució de tots els motors que intervenen en el sistema. La figura següent ajudarà a entendre el concepte.

```
102  word motors[2][9]={  
103      {9,2,3,4,8,5,6,7,14},  
104      {1,10,11,12,13,15,16,17,18}  
105  };  
106  
107  word *motorIDs;
```

Figura 38 – Estructura de dades per parametritzar el robot.

Com s'observa, l'estructura anomenada `motors[][]` indica que hi ha 2 robots al sistema i que cada un conté 9 servomotors. El que realitzen les línies 561 i 565 de la figura 37 és aplicar un offset a aquesta estructura i assignar l'adreça resultant a la variable `motorIDs`, que a partir d'aquest moment apuntarà al primer element del vector de motors del seu robot, i de cap altre més.

Aquesta parametrització permet jugar amb el concepte següent: definir les articulacions del robot de forma independent a aquest, per a poder-lo programar sense saber exactament quin robot hi ha connectat. De fet, aquest últim concepte és sobre el que es basa l'apartat actual.



```
44 #define BASE 0
45 #define ESPATLLA 1
46 #define COLZE 3
47 #define CANELL1 5
48 #define CANELL2 6
49 #define CANELL3 7
50 #define TOOL 8
```

Figura 39 – Definició de constants per cada articulació.

Fixem-nos primer que la numeració que segueixen no és consecutiva. Aquest fet és deliberat, ja que els identificadors de motor que estan a la posició 1 i 2 del vector `motorIDs` es corresponen amb la mateixa articulació, l'ESPATLLA, mentre que els identificadors situats a les posicions 3 i 4, corresponen a l'articulació COLZE.

Durant el que queda de codi, per accedir a una articulació determinada es farà tot utilitzant l'ús de les constants definides, de forma que la traducció d'una articulació al seu motor (o motors) es farà transparent a l'usuari, de la següent manera:

```
setMotor(motorIDs[CONST_ART], pos, vel);
```

D'aquesta manera, es manté un registre de tots els motors que hi ha al sistema, i paral·lelament es permet accedir a les articulacions sense tenir en compte quin robot està connectat.

Com s'ha anat observant al llarg dels punts anteriors, no es fa ús dels identificadors específics de cap motor ni de cap robot durant l'execució del codi de la controladora. Aquest és el resultat de la programació d'un firmware parametrizable i pensat per a ser executat en més d'un robot diferent.

## **7.6 Modes de funcionament del sistema**

Un cop analitzades totes les funcionalitats específiques del firmware controlador dels robots, tot seguit s'especificaran els quatre modes de funcionament que hi ha implementats. Tots els modes que s'han dut a terme estan pensats per donar assistència a les possibles necessitats del personal d'assistència en un quiròfan.

### **7.6.1 Mode de moviment teleoperat**

Aquest és el mode més elemental de tots. Permet la teleoperació dels robots a partir de l'ús de la botonera de la controladora i/o de l'aplicació executada a l'ordinador central. Seleccionant l'articulació concreta que es vol teleoperar, aquesta pot ser posicionada de forma incremental des de la seva posició actual fins a qualsevol punt, dintre dels límits físics del servomotor.

Es va decidir incorporar aquest mode al projecte al tractar-se del mode més elemental però també del més indicat per resoldre petits desajustos en el posicionament automàtic dels robots.

### **7.6.2 Mode de posicionament gestual del robot.**

Aquest és, potser, un dels modes de funcionament que més utilitat immediata pot arribar a tenir. El seu funcionament cerca la possibilitat de posicionar gestualment –és a dir, fent servir les mans del personal assistent- els dos braços robòtics en la posició desitjada pel cirurgià o pel personal assistent, i que el robot mantingui la posició que se li ha indicat. Els usos poden ser molt amplis, ja que doten al cirurgià/na d'un parell addicional de mans per a la subjecció d'instrumental o per la il·luminació de punts concrets de la taula d'operacions.

Aquest mode està implementat de la següent forma: Quan és assignada una posició al robot, aquest aplica un parell motor a cada un dels servomotors que

el conformen, per tal que aquests assoleixin i mantinguin la seva posició tot i possibles pertorbacions externes, o per tal de contrarestar el pes mateix del robot. Aquest parell aplicat fa que el robot no sigui susceptible de ser manipulat com si d'un braç mòbil es tractés, i l'objectiu d'aquest mode de funcionament és precisament donar el control a l'equip assistent sobre quan es vol que el robot permeti posicionar-se manualment i quan no. D'aquesta manera, quan és necessari, es deshabilita el parell motor de tots els servomotors i es posiciona el braç on es desitgi, per tot seguit tornar a habilitar el parell motor dels servomotors.

### ***7.6.3 Mode de cerca i tria d'instrumental quirúrgic***

Aquest mode de funcionament és més complex que els dos anteriors. La seva finalitat és discernir entre diferents tipus d'instrumental i aplicar un tractament diferenciat per a cada un d'ells. L'anàlisi d'aquest mode és interessant ja que hi participen els dos robots –funcionalitat cooperativa –, i es fa ús de la xarxa sense fils per a la comunicació i la coordinació entre l'ordinador i els dos braços robòtics.

Partint d'una zona delimitada d'entrada d'instrumental, el sistema és capaç de detectar les marques de colors que caracteritzen cada un dels instruments de treball, anar-los a buscar i transferir-los a l'altre robot per tal que en faci un tractament o un altre, segons l'instrument que sigui.

Tot seguit es presenta un cronograma on es pot veure el transcurs d'aquest mode de funcionament, i després s'analitzarà pas per pas.

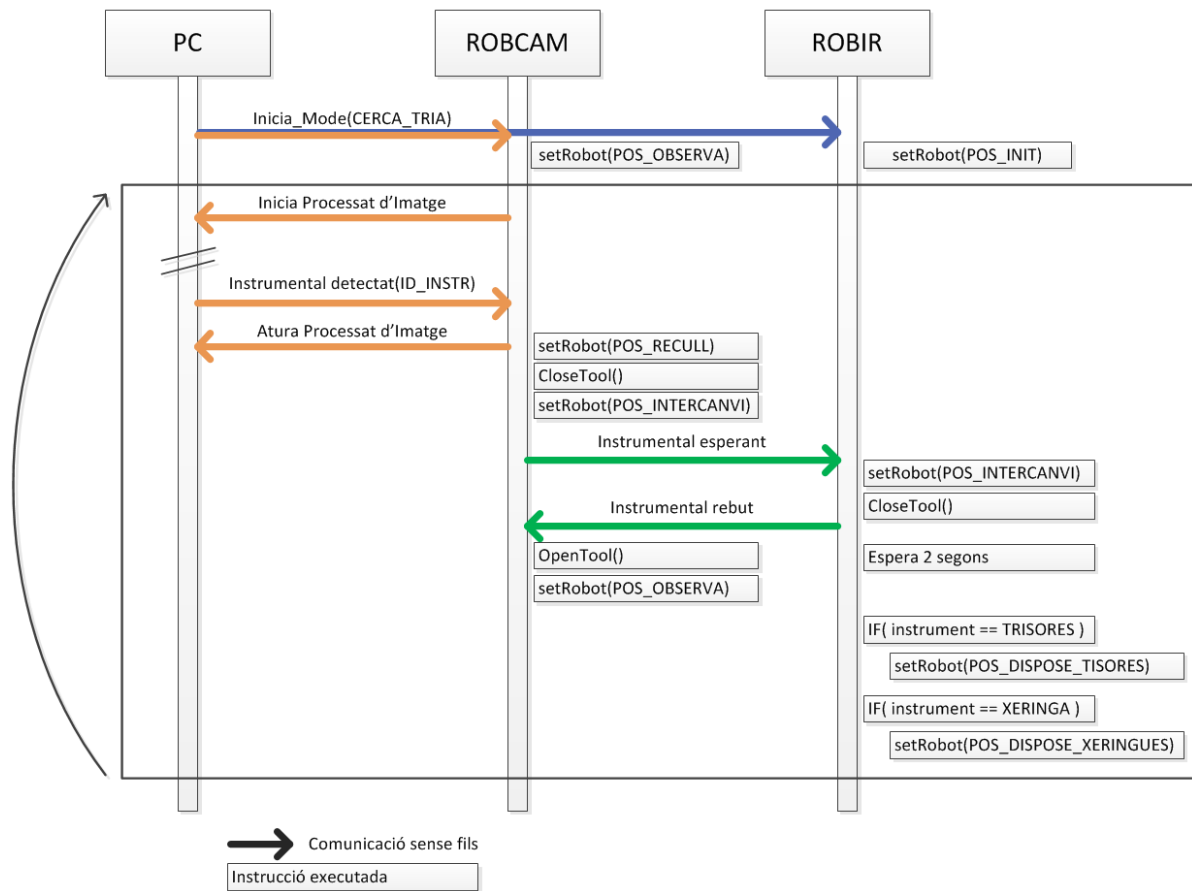


Figura 40 – Cronograma del mode de funcionament de cerca i tria.

Algunes observacions:

1. La comanda `Inicia_mode()` es transmet en broadcast cap als dos robots. Degut a que s'utilitzen 2 bits per l'adreçament de dispositiu i només hi ha 2 robots + 1 ordinador, hem pogut destinar el "quart dispositiu" al broadcast.
2. Després que ROBCAM comuniqui al PC que ha d'iniciar el processat d'imatge, el temps que passa no és constant. La interpretació ha de ser en forma de 'quan el PC detecta una nova eina...' envia la comanda `Instrumental_detectat()`<sup>22</sup>.

<sup>22</sup> Es recorda que tot el que fa referència a la detecció d'instrumental es tractarà a l'apartat de l'aplicació d'usuari.

#### 7.6.4 Mode per ordenar instrumental mèdic a la taula d'instrumental

Aquest mode de funcionament es basa en el mateix principi de detecció d'instrumental que l'anterior, però el tractament que en fa proporciona una nova utilitat a la plataforma de proves.

El que s'ha tractat d'assolir en aquest mode és capacitar a la plataforma de proves per ordenar tota una bateria d'instrumental mèdic, en funció de com estigui identificat (colors) i on es vulgui posicionar.

D'igual manera que en el mode anterior, tot seguit mostrarem un cronograma que farà molt més senzilla la seva interpretació.

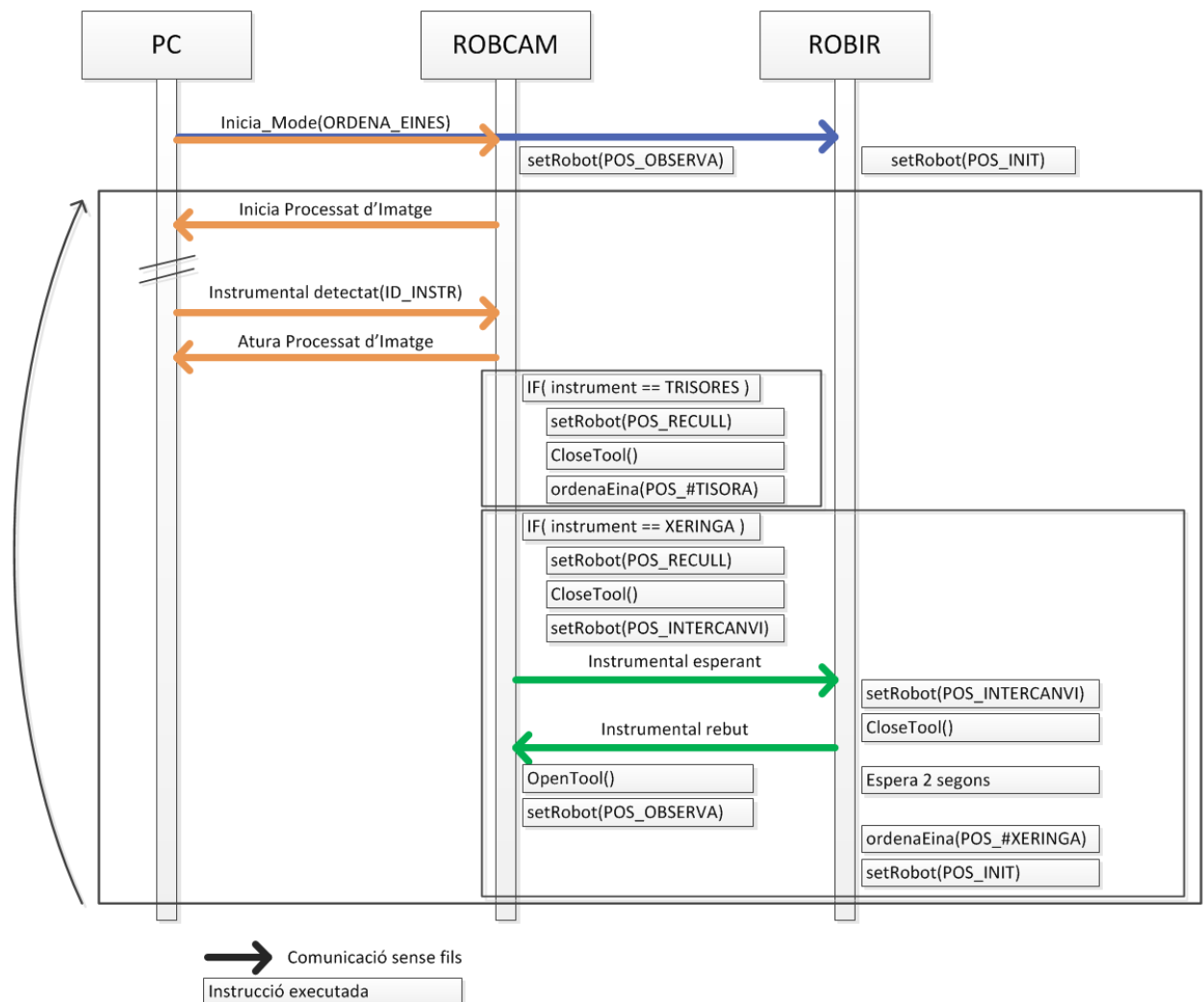


Figura 41 – Cronograma del mode d'ordenat d'instrumental mèdic.

Com es pot observar, hi ha similituds entre el mode anteriorment descrit, però la diferència fonamental és el tractament que es dona a l'eina detectada, ja que sigui una tisora o una xeringa (exemples que s'han fet servir durant les proves), marca el flux d'execució al ROBCAM.

Algunes consideracions:

1. Els valors POS\_#TISORA i POS\_#XERINGA no són constants, ja que depenen del nombre d'eines ja ordenades a la taula d'instrumental. Aquest recompte es porta a terme en temps real, mantenint actualitzat un comptador per cada tipus d'instrumental. D'aquesta manera es permet modificar la posició a la qual és transportat el nou instrumental que es vagi afegint a la taula i, a més a més, també es permet denegar la inserció de noves eines en cas de trobar-se la taula plena.
2. Fixar-se en que les instruccions dins dels quadres condicionals marquen execució d'un bloc de codi o un altre, mai els dos a l'hora.

### 7.7 Tecnologies i programari utilitzats

Per dur a terme la programació del codi i dels robots, s'han fet servir les següents tecnologies i programes:

- Llenguatge: C.
- Editor de textos per programar: Notepad++
- Compilador: GNU GCC WinAVR.
- Programació del robot: Robot Console.

Per finalitzar aquest apartat, es mostra un diagrama de connexió de l'ordinador amb la controladora i els servomotors. Aquesta configuració es fa servir per a programar el firmware.

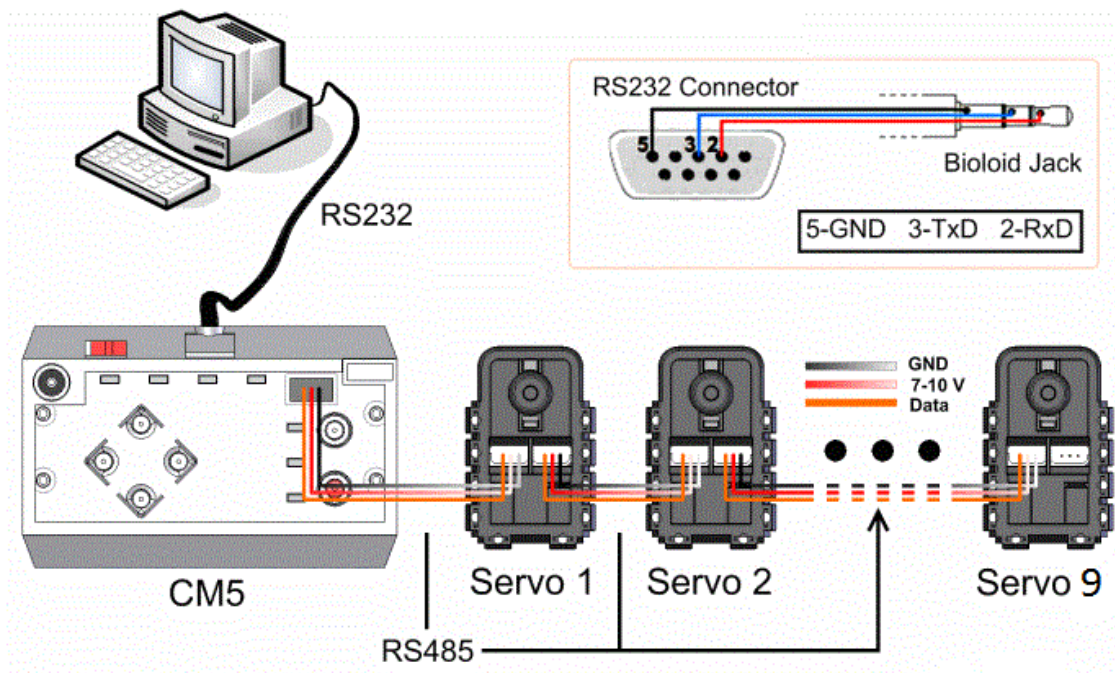


Figura 42 – Configuració del sistema quan es programa la controladora.

## 8 Programació de l'aplicació basada en PC

---

En aquest apartat es descriurà el procés de desenvolupament de l'aplicació basada en PC i les decisions que s'han pres. També es comentarà la seva utilitat pràctica en un entorn real de quiròfan.

Recordem que l'aplicació que s'ha dissenyat permetrà disposar d'una interfície d'usuari, per tal que entre el cirurgià i el seu equip existeixi una comunicació del sistema i cap al sistema.

A més a més, l'aplicació dotarà a la plataforma d'un sistema de visió per computador que ens permeti identificar instrumental mèdic. **Aquesta identificació serà a través de marques de colors a cada una de les famílies d'instrumental.** Això és així degut a les característiques de la càmera sense fils, que no permet la visió en 3D ni la identificació de formes complexes, degut a la seva resolució.

Pel que fa al processat d'imatge, una possible aproximació hagués estat la implementació de tot el sistema de processat d'imatge integrat al mateix firmware de la controladora, i fer un sistema 100% independent d'un ordinador extern. Aquesta aproximació, però, comportava l'ús de components no inclosos al kit de desenvolupament de Bioloid, i per tant de més difícil integració amb tot el sistema. Recordem que tant la càmera, com el receptor sense fils, així com les llibreries de programació, han estat subministrades pel fabricant i això, en general, és garantia d'un temps d'integració reduït.

Les llibreries de programació proporcionades pel fabricant permeten fer ús del tàndem càmera/receptor amb una certa facilitat, i proporcionen les rutines necessàries per a elaborar un algoritme de processat d'imatge prou complet per assolir els nostres objectius.

Abans de continuar, observem el diagrama de flux de l'aplicació desenvolupada.



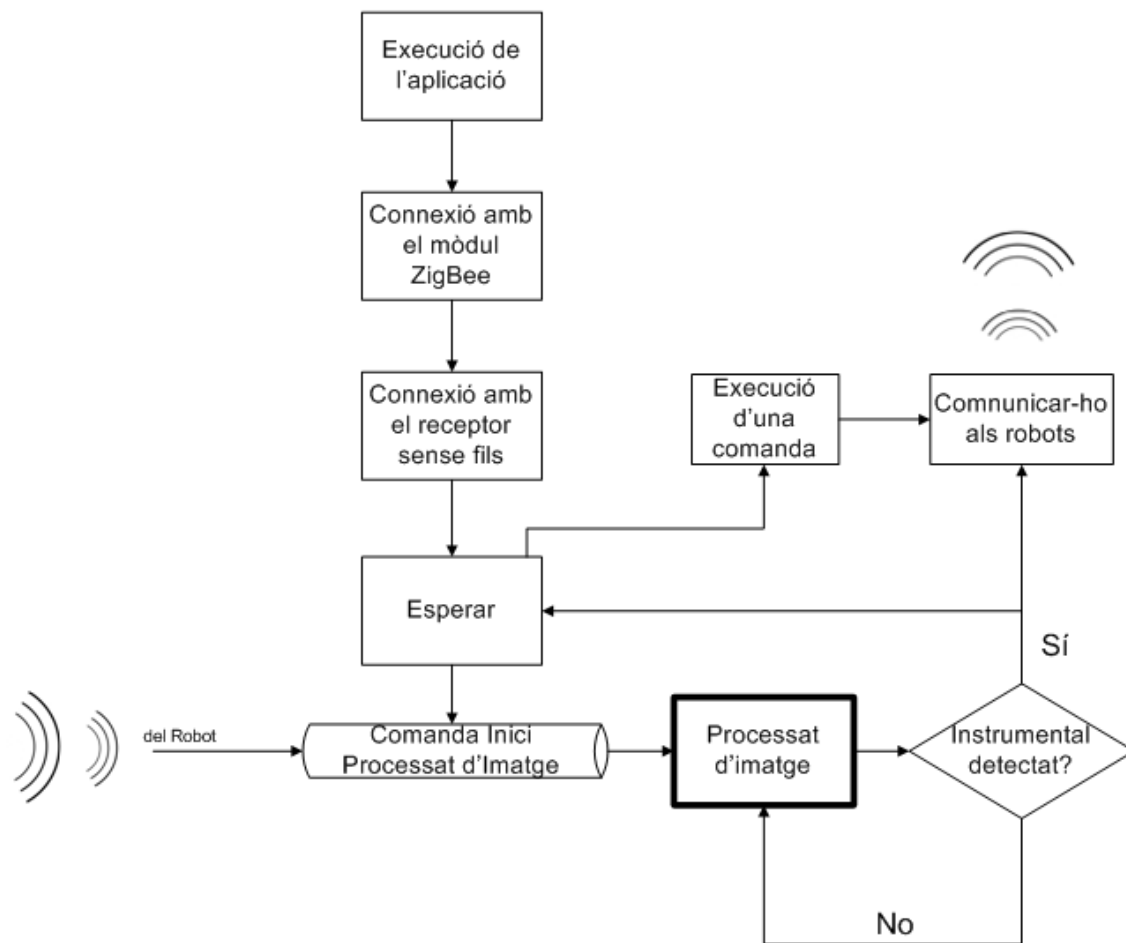


Figura 43 – Diagrama de flux de l'aplicació per a PC.

Com s'observa, s'ha ressaltat en negre el processat d'imatge. Precisament és en aquest algorisme on volem fer el primer incís. Ha quedat palès que aquest projecte no es tracta d'un projecte de visió per computador ja que no era l'objectiu principal, però sí que es pot dir que ha format part del gruix de reptes importants que s'han hagut de solucionar.

Que el sistema sigui capaç de diferenciar colors, separant així una mena d'instrumental d'un altre, ha estat una part molt important en el desenvolupament del projecte, i tot seguit es farà una breu explicació del mètode que s'ha seguit per assolir tal objectiu.

Abans de passar al tema del processat d'imatge, mostrem un diagrama de flux dels passos que es segueixen. Aquest diagrama es pot entendre com una ampliació del procés anomenat 'Processat d'imatge' de la figura 43.

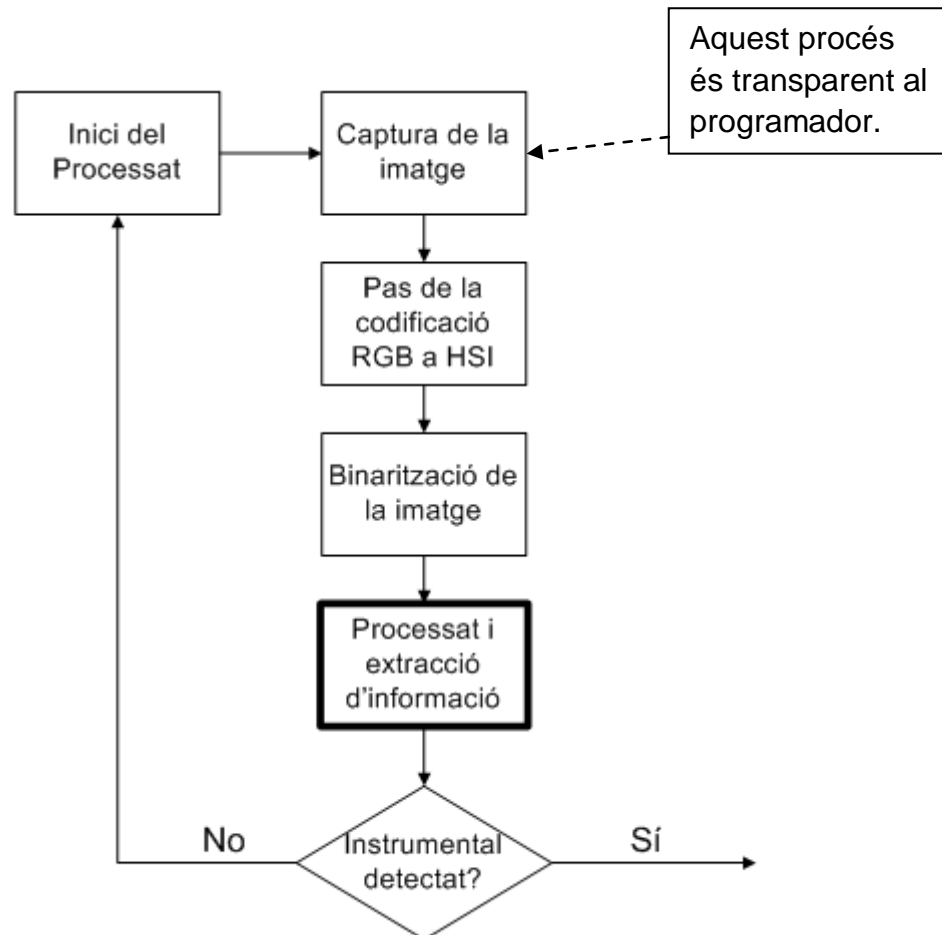


Figura 44 – Diagrama de flux del processat d'imatge.

Tot seguit es procedirà a fer una explicació detallada del processat d'imatge que efectua l'aplicació per a PC.

### 8.1 Pas de la codificació RGB a HSI.

Al sistema robòtic que ens ocupa, els components que han estat utilitzats provenen d'un kit de desenvolupament proporcionat pel fabricant. Aquest fet ha condicionat tant la qualitat de les imatges que es reben de la càmera, com la integració del sistema global. El que s'intenta fer en aquest apartat és explicar quina ha estat la raó de ser del sistema de visió, i com ha estat implementat. Per això es farà un breu repàs a tot el que és teoria del color, captura d'imatges, detecció d'objectes, etc.

Les imatges estan formades per una matriu de píxels –en el nostre cas, una matriu de 320x240-, i degut a les especificacions del material emprat en aquest projecte, aquestes arriben en el conegut format RGB –Red, Green, Blue-. Aquest format està representat per una tripla de valors hexadecimals de 8 bits cada un, és a dir, 8 bits pel vermell, 8 pel verd i 8 més pel blau. Representats en forma compacta es poden mostrar de la següent manera:

#551122

On el valor 0x55 representa els 8 bits del vermell (85 en decimal), el valor 0x11 representa els 8 bits del verd (17 en decimal) i el valor 0x22 representa els 8 bits del blau (34 en decimal). A la figura següent tenim una altra representació d'aquest color:

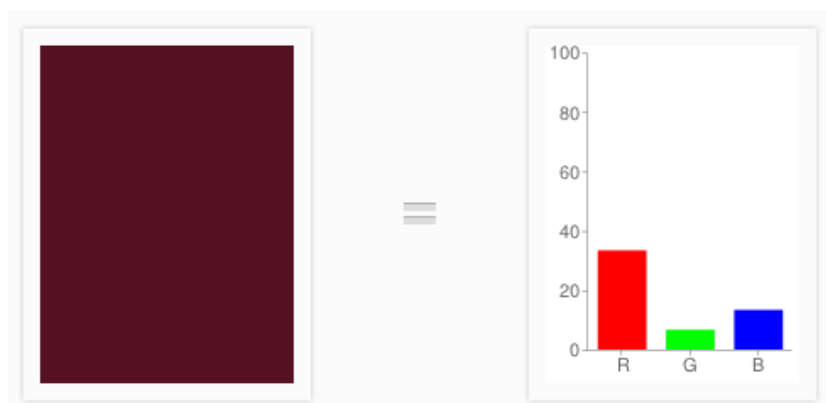


Figura 45 – Exemple de codificació del color #551122.  
La numeració de l'eix d'ordenades representa el % de color.

Aquesta codificació –la barreja dels tres colors primaris-, tot i que intuïtiva, no permet una programació ràpida i eficient d'un algorisme de processat d'imatge, ja que per un mateix valor de vermell es poden tenir 256 valors diferents de verd i 256 valors diferents més de blau, fent un total de  $256^2 = 65.535$  combinacions diferents. El mateix passa si fixem un valor de verd o de blau; la combinatòria no permet desenvolupar una aplicació eficient.

En canvi, si fem ús de la representació de color anomenada HSI –Hue, Saturation, Illumination- tot es simplifica més.

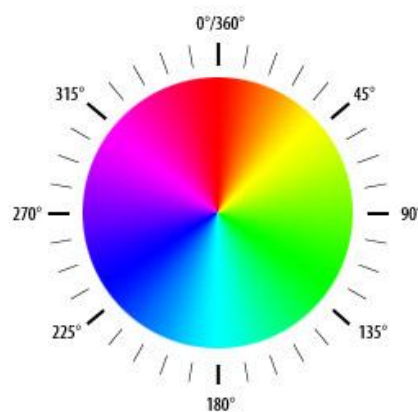


Figura 46 – Representació de l'escala de colors en format HSI.

Amb aquesta representació, discernir entre un color o un altre és tan senzill com avaluar el valor d'una sola variable, en aquest cas l'angle (Hue) determinat al disc HSI. Si el seu valor es situa als voltants de  $45^\circ$ , significa que el valor és groc. Si per un altre cas, el valor de l'angle és aproximadament  $220^\circ$ , es tracta de blau fosc.

Tot aquest procediment té com a resultat tres variables; la H –angle-, la S –saturació- i la I –il·luminació-. D'aquests tres paràmetres, amb la H i la S serà amb els que treballarem al següent apartat.

Afortunadament es va disposar de la implementació de la rutina de transformació de l'espai de colors RGB a l'espai de colors HSI, documentada a la memòria del projecte del Luis Botaya, pel que la tasca que centra el

processat d'imatge d'aquest projecte es centra en les decisions que pren el sistema a mesura que es van incorporant nous instruments de treball a la plataforma.

Tot seguit analitzarem el procediment corresponent al procés anomenat 'Binarització de la imatge' de la figura 44.

## 8.2 Binarització<sup>23</sup> de les imatges

Un cop implementat la transformació d'un espai de colors a un altre, encara queda un pas previ a poder començar a prendre decisions. Aquest pas es tracta de la binarització de la imatge. Aquest procés es farà assignant un valor entre 0 i 255 als colors que es vulgui identificar, essent 0 el negre i 255 el blanc, passant per una escala de 8 bits de grisos.

Partim d'una imatge representada amb l'espai de colors HSI i de la qual, per exemple, en volem extreure els píxels de color blau. Per tal de poder processar de forma més eficient aquesta imatge, el procediment que s'ha seguit és detectar els píxels que estan entre els valor que es consideren blau –recordem que el blau es troba al voltant del valor 220 del Hue de l'escala HSI- i assignar un valor entre 0 i 255 a la imatge resultat. El resultat es pot observar a la figura següent:

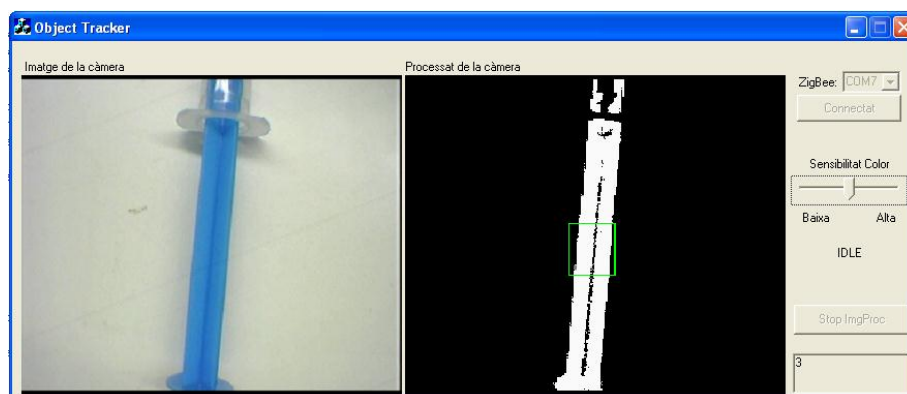


Figura 47 – Binarització de la imatge d'una xeringa.

<sup>23</sup> Conversió d'un valor real a un valor binari. En el nostre cas, però, es passarà d'una imatge de valors reals a una a escala de grisos, amb l'objectiu de detectar més instruments mèdics.

Cal observar que només es veuen representats els píxels que són de color blau; més concretament del blau que el programa té parametritzat. Si es volgués identificar més d'un color, la imatge binaritzada mostraria els dos objectes identificats, representats en diferents tonalitats de gris.

Tot seguit es mostra una captura de pantalla del codi que du a terme la binarització, i a continuació s'explica:

```
/**
    [250]->blau
    [245]->groc
**/

if( S > 80 && S < 175 ){                                //SATURACIO CORRECTE
    if( H >= (180-range) && H <= (180+range) ){          //DETECTEM BLAU
        pBinary[index]=250;
    }
    else if( H <= 60+range || H >= (60-range) ){        //DETECTEM GROC
        pBinary[index]=245;
    }
    else{
        pBinary[index]=0;
    }
} else{
    pBinary[index] = 0;
}
```

Figura 48 – Implementació de la binarització d'un píxel.

Primer de tot cal comentar la comprovació que el nivell de saturació estigui entre un interval acceptable, ja que depenent d'aquest valor, el color corresponent al píxel que s'estigui processant no serà susceptible de ser tractat. Un color blau molt o poc saturat pot no assemblar-se al blau ideal que l'aplicació ha d'identificar, i per tant és correcte descartar-lo.

A continuació observem que es controla el valor de la variable H, corresponent al Hue (angle) del disc que s'ha vist a la figura 46. Si el seu valor està al voltant d'un llindar determinat, marcat pel valor teòric del color que estem buscant i el valor d'un llindar *range*, llavors es considera que el píxel en procés es correspon al cas que estem tractant, assignant un valor arbitrari de l'escala de

grisos a la posició emmagatzemada a la variable "*index*" –representant de la fila i la columna- del vector representatiu de la imatge binària, variable "*pBinary*".

Aquest procediment, dut a terme per a cada píxel de la imatge i discernint entre els colors que es vulgui identificar, dóna lloc a la imatge binaritzada que es veu a la dreta de la figura 47.

Com a apunt final, destacarem la limitació del nombre d'eines que és capaç de diferenciar el sistema. El nombre teòric són 256, ja que el disc HSI està dividit utilitzant 8 bits (256), però el nombre real és menor, ja que per la identificació d'un color es fa servir un rang de valors que pot ser estret, aproximadament de 20, o bé més eixamplat, d'aproximadament 60. Aquest ventall és degut principalment a la definició de la càmera i a les condicions de llum.

### **8.3 Algorisme de processat d'imatge**

Un cop disposem de la imatge en escala de grisos amb els objectes desitjats representats, és hora de prendre les decisions pertinents que ens permeten identificar quin instrument hi ha a la plataforma, i què hem de fer amb ell.

Un dels objectius del projecte era identificar un nombre finit però suficient d'instruments quirúrgics, així com el de construir un sistema escalable. Seguint aquestes dues consignes s'ha dissenyat el sistema per tal que pugui identificar un màxim teòric de 256 eines, emmagatzemant la seva informació en tota una sèrie de variables associades, indistintament de 'quina' eina s'estigui tractant. Aquest concepte dota al sistema de l'abstracció necessària per gestionar tot tipus d'instrumental, sense parlar de cap en concret.

Pas previ a l'anàlisi del codi, es mostrarà un digrama de flux amb els passos que es duen a terme en aquest procés.

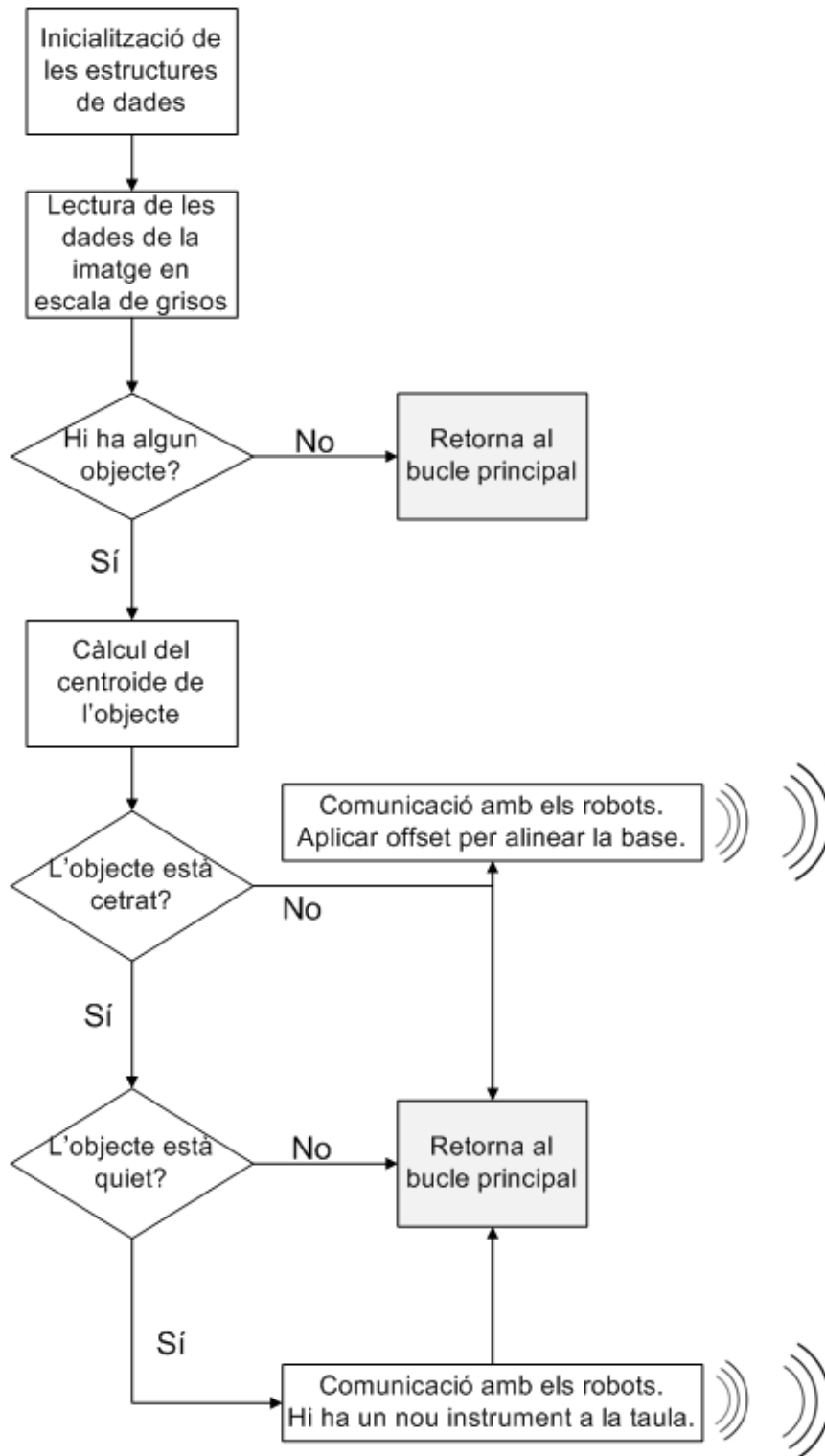


Figura 49 – Diagrama de flux del procés de detecció d'un objecte.



I ara, tot seguit, es mostra la inicialització de les estructures de dades necessàries de l'algorisme de processat d'imatge.

```
void _ProcessaBN(unsigned char* pBinary, int* cenX, int* cenY){  
    int cX, cY;  
    int x, y, index;  
    int i;  
  
    unsigned int aCount[256];  
    unsigned int aSumX[256];  
    unsigned int aSumY[256];  
    unsigned int aMaxX[256];  
    unsigned int aMaxY[256];  
    unsigned int aMinX[256];  
    unsigned int aMinY[256];  
  
    for (i=0; i<256; i++){  
        aCount[i]=0;  
        aSumX[i]=0;  
        aSumY[i]=0;  
        aMaxX[i]=0;  
        aMaxY[i]=0;  
        aMinX[i]=0;  
        aMinY[i]=0;  
    }  
}
```

Figura 50 – Inicialització de les estructures de dades del processat d'imatge.

El primer que cal comentar són els paràmetres d'entrada. El primer, `char *pBinary`, es correspon al punter a la matriu que conté la imatge en escala de grisos. Recorrent aquesta matriu serà com extraurem la informació de la imatge que la càmera està veient. Els altres dos paràmetres es corresponen a l'adreça de memòria de les variables utilitzades a la rutina principal per determinar el centre de masses de l'instrument. No tenen rellevància per l'explicació, ja que el seu ús està limitat a pintar el quadre verd que es pot observar a la figura 47.

Els vectors definits a continuació, són:

- `aCount`: Manté el recompte de píxels de cada color.
- `aSumX`: Manté el sumatori dels valors de la component X de cada color.
- `aSumY`: Manté el sumatori dels valors de la component Y de cada color.
- `aMaxX`: Manté el valor màxim de la component X de cada color.

- aMaxY: Manté el valor màxim de la component Y de cada color
- aMinX: Manté el valor mínim de la component X de cada color.
- aMinY: Manté el valor mínim de la component Y de cada color.

Els vectors aSumX i aSumY es fan servir posteriorment per determinar el centre de l'eina a partir del mètode del centroide<sup>24</sup>.

Com es pot observar, mantenir la informació de tot el possible instrumental té un cost espacial aproximat de 7 vectors x 256 posicions x 4 bytes/int = 7.168 Bytes = 7 KBytes. Aquest fet pot limitar la possible decisió futura de traslladar al robot la tasca del processat d'imatge.

La següent fase del processat és llegir les dades de la imatge i carregar-les a les estructures de dades prèviament inicialitzades. A partir d'aquesta informació serà quan prendrem les decisions futures. Tot seguit observem la implementació proposada.

```
// Recorrem la imatge per files i columnes
for( y=0; y<VP_HEIGHT; y++ ){           //FILES
    for( x=0; x<VP_WIDTH; x++ ){         //COLUMNS
        index = VP_WIDTH*y + x;

        int color=pBinary[index];

        if(color>0){
            aCount[ color ]++;

            aSumX[ color ] += x;
            aSumY[ color ] += y;

            if(x<aMinX[ color ]) aMinX[ color ]=x;
            if(x>aMaxX[ color ]) aMaxX[ color ]=x;

            if(y<aMinY[ color ]) aMinY[ color ]=y;
            if(y>aMaxY[ color ]) aMaxY[ color ]=y;
        }
    }
}
```

Figura 51 – Implementació de l'obtenció de dades de la imatge binaritzada.

<sup>24</sup> <http://en.wikipedia.org/wiki/Centroid>

A partir de la posició del píxel que s'està avaluant, es procedeix a actualitzar les estructures de dades concretes, com el comptador de píxels d'aquest color determinat, la posició màxima i mínima, etc. Recordem que el fet que un píxel sigui d'un color determinat es tradueix en que la càmera està veient un píxel d'aquest color i això potencialment ens indica que hi ha un instrument determinat.

A partir d'aquí el sistema està en disposició d'analitzar aquestes dades i prendre la decisió de comunicar als braços robòtics la detecció d'un instrument nou, la necessitat de corregir la posició d'observació, etc.

Cerca del primer color amb +400 píxels a la imatge.

```
for(i=255; i>0; i=i-5){
    if(aCount[i] > 400) break;
}
```

Càlcul del centre de l'objecte mitjançant el mètode del centroide.

```
if(i>0){
    cX= (int)(aSumX[i] / aCount[i]);
    cY= (int)(VP_HEIGHT-(aSumY[i] / aCount[i]));
```

```
int diff=(VP_WIDTH/2)-cX;
SetDlgItemInt(pThis->GetSafeHwnd(), IDC_DEBUG, diff, false );
```

Control de la posició de l'objecte.

```
if(abs(diff)>40){
    short dades;
    if(diff<20) dades=89;
    else dades=25;
```

Correcció de la posició del robot.

```
ZigSend(CMD_MOVEARTREL, ID_ROBCAM, dades);
}else{
    *cenX=cX;
    *cenY=cY;
```

Control que l'objecte estigui quiet mitjançant el processat de LLINDAR\_CAPTURES imatges.

```
if( (abs(cX-oldeX)<10) && (abs(cY-oldeY)<10) ){
    framesOK++;
}else{
    oldeX=cX;
    oldeY=cY;
    framesOK=0;
}
if(framesOK==LLINDAR_CAPTURES){
    ZigSend(CMD_NEW_TOOL, ID_ROBCAM, i);
    framesOK=0;
    timer_flag=0;
}
```

Comunicació amb el robot per indicar que hi ha una nova eina esperant.

Figura 52 – Implementació del processat de les estructures de dades amb informació sobre la imatge binaritzada.

Durant el desenvolupament d'aquest codi, es va decidir incorporar una funcionalitat de correcció de posicionament del robot segons l'objecte visible per la càmera estigüés més a la dreta o més a l'esquerra del centre de visió. Això permet disposar l'eina de treball en una zona indeterminada dins la zona d'entrada, i que el sistema s'encarregui de fer la resta, corregint la posició del robot i recollint l'objecte des de la nova posició.

Aquesta funcionalitat s'ha assolit mitjançant el càlcul de la posició central de l'objecte i pel fet de poder enviar comandes concretes al sistema robòtic.

A més a més, l'algorisme implementat té la possibilitat de saber si l'objecte està quiet o no. Aquesta funcionalitat s'ha implementat per evitar mal funcionaments derivats de capturar imatges massa de pressa (en el nostre cas, quatre cops per segon), com ara la detecció precoç d'instrumental quan aquest encara no estigüés situat definitivament a la zona d'entrada.

El sistema utilitzat és, a partir del centre calculat de l'objecte visible per la càmera, calcular si aquest centre s'ha mogut més o menys de 10 píxels respecte la captura anterior. Si aquest fet es repeteix LLINDAR\_CAPTUES vegades (que per a la demostració ha estat configurat a 4), el sistema decideix que hi ha un instrument mèdic nou a la zona d'entrada, i envia la comanda corresponent als robots, activant tot el mecanisme associat al mode de treball que s'estigui executant en aquell moment.

#### **8.4 Interfície gràfica d'usuari**

Un cop vist el funcionament de l'algorisme de processat d'imatge, i s'ha detallat com es procedeix a la identificació de colors, d'instrumental, etc. tot seguit es farà un breu comentari sobre la interfície gràfica d'usuari.

Aquesta part de l'aplicació, la part més visual, permetrà al cirurgià i al seu equip, disposar d'una sèrie d'eines i facilitats per interactuar amb la plataforma robòtica.

Com s'ha vist al principi d'aquest document, el cirurgià pot accedir als robots directament a través de la botonera situada a la seva controladora, però això no és del tot viable en un entorn de quiròfan, ja que els botons són petits, susceptibles d'embrutar-se etc.

Per això s'ha dissenyat una interfície gràfica d'usuari (GUI) que permetrà al cirurgià o al seu equip interactuar amb el sistema de forma còmode i intuïtiva.

Primer de tot veiem una captura de pantalla.

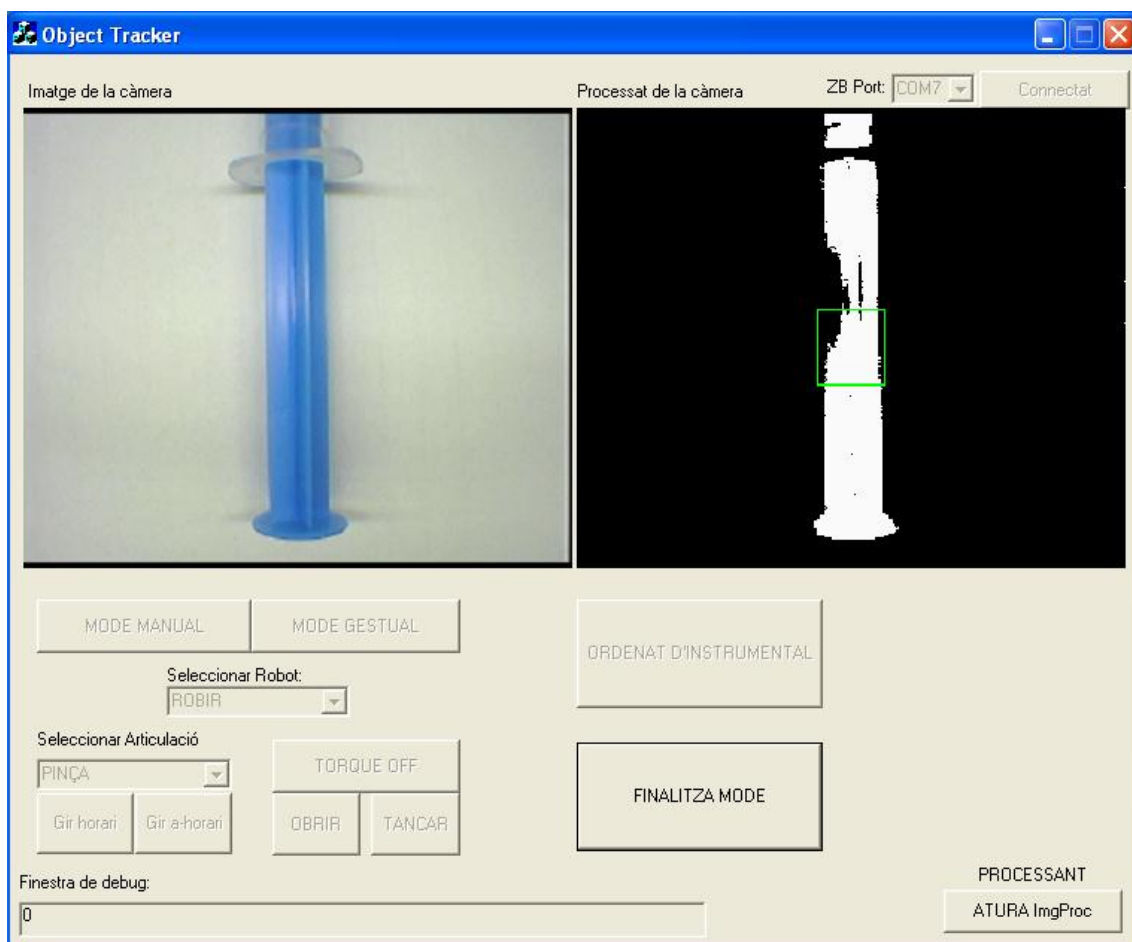


Figura 53 – GUI del sistema, en un moment de captura d'imatges.

Podem observar que, dins d'aquesta GUI hi ha dos apartats clarament diferenciats. El primer es situa a la part superior, i és on es pot observar la visió

que obté la càmera, juntament amb la conversió a l'escala de grisos que s'ha comentat a l'apartat anterior.

La part posterior de l'aplicació permet la interacció pròpiament dita. Podem observar que es pot canviar de mode de funcionament, activar i desactivar el parell motor de les articulacions del robot (recordem el mode de funcionament gestual), manipular les articulacions dels robots de forma teleoperada, etc.

L'escenari ideal és la manipulació d'aquesta interfície a través d'un monitor tàctil, de forma que el cirurgià ràpidament pugui canviar modes de funcionament, manipular el sistema, consultar l'estat dels robots, etc, amb un simple toc tàctil.

Recordem que, igual que la comunicació robot – robot, les dades que envia l'ordinador als robots s'efectuen mitjanant el mòdul de comunicacions sense fils ZigBee, i amb la implementació descrita a l'apartat 7.3.

A mode d'exemple, es mostra el fragment de codi que indica als robots que s'ha canviat al mode de posicionament gestual, que s'inicia fent clic sobre el botó etiquetat com a 'MODE GESTUAL'.

```
if(modeGestual==false){  
    modeGestual=true;  
    m_Button_ModeGestual.SetWindowText( "MODE GESTUAL. ATURAR." );  
    m_Button_ModeManual.EnableWindow( false );  
    m_Button_Sort.EnableWindow( false );  
    m_Button_Dispose.EnableWindow( false );  
  
    m_Combo_Robs.EnableWindow( true );  
    m_Button_ToolOp.EnableWindow( TRUE );  
    m_Button_ToolCl.EnableWindow( TRUE );  
    m_Button_TorqOn.EnableWindow( TRUE );  
    m_Button_TorqOff.EnableWindow( TRUE );  
    ZigSend(CMD_CHG_SYSMODE, ID_ROBCAM, MODE_GESTUAL);  
    ZigSend(CMD_CHG_SYSMODE, ID_ROBIR, MODE_GESTUAL);  
}  
else{
```

} Gestió dels altres elements visuals de la GUI.

} Comunicació amb els dos robots.

Figura 54 – Fragment de codi amb l'inici del mode de posicionament gestual.

## 9 Conclusions

Arribats a aquest punt, es procedeix a fer una valoració final de l'assoliment dels objectius fixats, quin ha estat el seu desenvolupament i la projecció que les tasques fetes poden tenir de cara a un quiròfan real.

Dels objectius marcats al principi del projecte se n'han complert els següents.

- Detectar instrumental quirúrgic en una zona acotada d'inserció.
- Manipular aquest instrumental quirúrgic.
- Fer-ho de forma cooperativa en cas de ser necessari, podent intervenir un o ambos robots de la plataforma.
- Implementar un sistema de comunicació fiable que sigui independent de l'ordinador de central i permeti la comunicació robot-robot i PC-robot.
- Desenvolupar una interfície d'usuari que permeti interactuar amb el sistema, tant en l'enviament de comandes com en la recepció de dades del sistema.
- Portar el registre de l'instrumental situat a la taula d'instrumental.

En canvi, no s'han pogut assolir els següents.

- Detectar instrumental quirúrgic fora de la zona acotada.
- Obtenir una GUI potent que ens permeti saber l'estat del sistema en qualsevol moment.
- En general, desenvolupar un sistema flexible i més versàtil.

Com s'ha comentat amb anterioritat, s'han patit alguns contratemps de caire tècnic, com ara la baixa resolució de la càmera o la poca fiabilitat del posicionament dels robots, fets que han condicionat en bona part el desenvolupament del projecte.

Tot i així s'ha assolit un sistema capaç de manipular instrumental mèdic, transportar-lo d'un braç robòtic a un altre, i que resol problemes del dia a dia en un quiròfan, com ara el recompte d'eines, la gestió d'aquestes abans i durant una intervenció, l'ús dels robots com un parell assistents de mans addicionals, etc.

Paral·lelament a aquests objectius assolits, s'ha aconseguit posar la base al possible desenvolupament posterior d'aquest mateix projecte, ja que les rutines i funcions programades estan pensades per a ser reutilitzades amb el menor temps d'adaptació possibles.

Com s'ha descrit en l'apartat de la programació del firmware, aquest s'ha programat fent una abstracció estudiada i curiosa del nivell del hardware del robot al nivell del firmware i de les llibreries de programació. Això ha de ser considerat com un èxit que, tot i no satisfer necessitats immediates en un quiròfan, permet que aquest projecte tingui continuïtat temporal.

La interfície d'usuari permet les funcions bàsiques de monitoreig del processat d'imatge i d'interacció usuari/sistema.



## 10 Anàlisi del cost econòmic

Tot seguit es realitza un anàlisi del cost econòmic del projecte actual. S'hi poden diferenciar dos apartats, material i recursos humans.

Entenem com a material tot allò corresponent al hardware i al software que ha estat necessari pel desenvolupament del projecte.

Material	Preu (€)
Kit de desenvolupament Bioloid + Software Bioloid RoboPlus	715
Fusta i metacrilat per la base de la plataforma de proves.	20
Components addicionals. Mòduls ZigBee i ferreteria.	100
Amortització de l'equip informàtic utilitzat durant 5 mesos.	125
Microsoft Office + Microsoft Visual Studio 2007.	0 <sup>25</sup>
<b>TOTAL:</b>	<b>960</b>

Entenem com a recursos humans el cost en hores per desenvolupar una tasca determinada. La variació en el cost/hora és deguda a que s'ha cregut convenient assignar les tasques a diferents tipus de professionals, descrits a continuació:

- **Tècnic:** Persona dedicada a desenvolupar les tasques mecàniques com la construcció de la base de la plataforma, integrar i documentar.
- **Programador:** Desenvoluparà la tasca de programar el firmware i l'aplicació per a PC tal i com especifiqui l'analista.
- **Analista:** Encarregat de realitzar l'estudi previ i el disseny de la solució final.

<sup>25</sup> S'ha fet ús de la llicència educativa de la Universitat Politècnica de Catalunya.

A la taula següent s'especifica la distribució de tasques i la seva càrrega en hores.

<b>Recursos humans</b>				
<b>Tasca realitzada</b>	<b>Hores</b>	<b>Treballador</b>	<b>€/hora</b>	<b>Preu (€)</b>
Anàlisi del material existent.	30	Analista	40	1.200
Disseny i construcció de la base de la plataforma de proves i de la segona unitat robòtica.	60	Tècnic	15	900
Anàlisis i disseny de la solució final.	75	Analista	40	3.000
Programació del firmware dels robots	150	Programador	30	4.500
Programació de l'aplicació basada en PC.	100	Programador	30	3.000
Integració de tot el sistema.	100	Tècnic	15	1.500
Documentació	120	Tècnic	15	1.800
Provar el sistema.	20	Tècnic	15	300
<b>TOTAL:</b>	<b>655</b>			<b>16.200</b>

El pressupost total del projecte, doncs, s'estima en material + recursos humans:

$$960 + 16.200 =$$

<b>17.160€</b>
----------------

## 11 Treball futur

Centrarem la descripció del treball futur en els següents punts.

### **11.1 Sistema de visió per computador**

Un dels punts clau del sistema global és la capacitat de la visió per computador. És evident que la identificació de l'instrumental, la seva ubicació a la taula d'operacions i la gestió de la seva posició en temps real són capacitats necessàries per a un sistema competent i capaç de treballar en un quiròfan.

Actualment, el sistema que s'ha dissenyat té instal·lada una càmera de baixa resolució a l'extrem d'un dels robots, i això comporta els següents problemes:

1. Sistema de referència mòbil. A mesura que el robot es mou, es pot perdre de vista la globalitat de tot l'entorn.
2. Baixa resolució. No permet la identificació d'instrumental a través de la seva forma, només ho permet a través de marques de colors.
3. Pèrdua de precisió per l'acumulació dels errors de posicionament derivats de la fragilitat de la base del robot.

Es proposen les següents mesures de cara a l'ampliació d'aquest projecte.

1. Dissenyar i construir un sistema de visió fix, instal·lat en una estructura que envolti tot el llit d'operacions. D'aquesta manera eliminem els errors de posicionament que s'arrossegueu quan la càmera està instal·lada en un dels robots i també permet al sistema tenir una visió global de la situació.
2. Dotar aquest sistema d'un mínim de dues càmeres d'alta resolució que permetin tant la estereovisió –determinar la profunditat de la imatge- com la identificació de l'instrumental a partir de la seva forma –fent ús de patrons-

### **11.2 Braços robòtics**

Dotar al sistema de la fiabilitat de braços robòtics industrials és una de les possibles millores que es poden aplicar al sistema. Actualment els braços que hi ha instal·lats no permeten un posicionament fiable, degut al pes que han de suportar les articulacions i a la limitada potència del servomotors.

Així doncs, substituir els dos braços actuals per uns de qualitat superior és un objectiu a assolir en futurs projectes, limitat únicament per motius econòmics, ja que la morfologia que segueix un robot industrial i la que segueix un robot de desenvolupament són bàsicament les mateixes.

Paral·lelament a substituir els braços actuals per uns de més qualitat, una línia a seguir seria el desenvolupament d'un sistema de parametrització automàtica del sistema. Tot allò descrit a l'apartat 7.5 referent a l'ús d'un sol codi per a N robots, es podria veure millorat amb la incorporació d'un sistema, p. ex. basat en XML, que permetés introduir les dades necessàries de parametrització de forma intuïtiva i gestionés la modificació del codi per tal de que un cop compilat i carregat als robots, funcionés de forma correcta.

### **11.3 Interfície Gràfica d'Usuari**

Una altra tasca ha estat la interfície gràfica d'usuari –GUI-. Durant el visionat dels vídeos de demostració del sistema robòtic *Penelope*, es poden observar algunes imatges que presenten una GUI potent, amb la representació de tota la informació sobre l'instrumental que hi ha a la taula, l'estat del sistema, i sobretot que transmet a l'usuari la sensació d'estar treballant amb un sistema intel·ligent, robust i que és capaç de gestionar problemes sense la intervenció de cap assistent.

Un objectiu per a desenvolupar en un futur projecte passaria per programar una GUI igual o més potent, que recollís tota la informació provinent del sistema de visió perfeccionat que s'ha comentat al punt 11.1 i, juntament amb la informació

de l'estat dels robots, ho traduís tot a una pantalla on a simple cop d'ull qualsevol assistent tingués el coneixement global de la situació.

A més a més, continuant el que es comenta a l'apartat 8.4, una possible línia a seguir seria la de evolucionar la GUI cap a una interfície tàtil, per a major comoditat del cirurgià, o fins i tot que fos capaç de respondre comandes per veu. Així només caldria anomenar l'instrument que es necessiti, i el sistema seria capaç de saber de quin es tracta, cercar si es troba disponible, i gestionar-lo de forma eficient.

## 12 Bibliografia

---

Robotis Homepage: [http://www.robotis.com/xe/bioloid\\_en](http://www.robotis.com/xe/bioloid_en)

Bioloid. *Expert Manual*.

Bioloid. *User Guide*.

Botaya, L. (2011). *Sistema robòtic per assistència puntual en quiròfans*.  
Barcelona.

Microsoft. *Visual C++ Programmer's Guide*. Consultat el 2012, a  
[http://msdn.microsoft.com/en-us/library/aa733974\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa733974(v=vs.60).aspx)

*Wikipedia*. Consultat el 2012, a <http://www.wikipedia.org>